



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA

**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE
SISTEMAS**

“Sistema web utilizando el protocolo de Manchester para el proceso de
emergencia médica en la Clínica Sanna San Borja”

TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:

Ingeniero de Sistemas

AUTORES:

Br. Garcia Del Aguila, Oscar (ORCID: 0000-0003-4079-1583)

Br. Panizo Gonzales, Lincoln King (ORCID: 0000-0001-7494-4504)

ASESOR:

Mg. Gálvez Tapia, Orleans Moisés (ORCID: 0000-0002-4352-9495)

LÍNEA DE INVESTIGACIÓN:

Sistemas de Información y Comunicaciones

LIMA – PERÚ

2019

DEDICATORIA

A mis padres Mirtila y Clemente, a mis hermanos Peggy y Clement, por su apoyo incondicional y su ejemplo de esfuerzo, coraje y valentía. Al Dr. Adilio Christian Ordoñez Pérez y al Mg. Orleans Moisés Gálvez Tapia, por apoyarnos en sus asesorías en el desarrollo de esta tesis.

Lincoln King Panizo Gonzales

A mis padres Rosa y Oscar, a mis hermanos Leonela y Miguel, a mi novia Miluska, por su apoyo incondicional y su ejemplo de esfuerzo, coraje y valentía.

Al Dr. Adilio Christian Ordoñez Pérez y al Mg. Orleans Moisés Gálvez Tapia, por apoyarnos en sus asesorías en el desarrollo de esta tesis.

Oscar Garcia Del Aguila

AGRADECIMIENTO

A Dios, por darme la fuerza que necesitaba para seguir adelante. A mi madre Mirtila y mi padre Clemente, por su apoyo.

Lincoln King Panizo Gonzales

A Dios, por darme la fuerza que necesitaba para seguir adelante. A mi padre Oscar y mi madre Rosa, por su apoyo.

Oscar Garcia Del Aguila

PÁGINA DEL JURADO

DECLARATORIA DE AUTENTICIDAD

DECLARACIÓN DE AUTENTICIDAD

Yo, Oscar García del Águila y Lincoln King Panizo Gonzales, estudiantes de la Escuela Profesional de Ingeniería de Sistemas de la Universidad César Vallejo, sede Lima Norte; presento el trabajo académico titulado "SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO DE EMERGENCIA MEDICA EN LA CLINICA SANNA SAN BORJA", para la obtención del título profesional de Ingeniería de Sistemas.

Por lo tanto, declaro lo siguiente:

He mencionado todas las fuentes, de acuerdo con lo establecido por las normas de elaboración de trabajos académicos

Este trabajo de investigación no ha sido previamente presentado completa ni parcialmente para la obtención de otro grado académico o título profesional.

Soy consciente de que mi trabajo puede ser revisado electrónicamente en búsqueda de plagios.

De encontrar uso de material intelectual ajeno sin el debido reconocimiento de su fuente o autor, me someto a las sanciones que determinan el procedimiento disciplinario.

Lima, 19 de diciembre del 2019



García del Águila Oscar

DNI N° 47772043



Panizo Gonzales Lincoln King

DNI N° 46760983

PRESENTACIÓN

Señores miembros del Jurado:

Dando cumplimiento a las normas establecidas en el Reglamento de Grados y Títulos sección de Pregrado de la Universidad César Vallejo para la experiencia curricular de Metodología de la Investigación Científica, presento el trabajo de investigación pre-experimental denominado: “SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO DE EMERGENCIA MEDICA EN LA CLINICA SANNA SAN BORJA”.

La investigación, tiene como propósito fundamental: Determinar la influencia del Sistema Web utilizando el protocolo de Manchester en el proceso de Emergencia Médica en la Clínica Sanna San Borja.

La presente investigación está dividida en siete capítulos:

En el primer capítulo se expone el planteamiento del problema: incluye formulación del problema, los objetivos, la hipótesis, la justificación, los antecedentes y la fundamentación científica. En el segundo capítulo, que contiene el marco metodológico sobre la investigación en la que se desarrolla el trabajo de campo de la variable de estudio, diseño, población y muestra, las técnicas e instrumentos de recolección de datos y los métodos de análisis. En el tercer capítulo corresponde a la interpretación de los resultados. En el cuarto capítulo trata de la discusión del trabajo de estudio. En el quinto capítulo se construye las conclusiones, en el sexto capítulo las recomendaciones y finalmente en el séptimo capítulo están las referencias bibliográficas.

Señores miembros del jurado espero que esta investigación sea evaluada y merezca su aprobación.

Índice

	Página
Cáratula.....	i
Dedicatoria	ii
Agradecimiento	iii
Página del jurado	iv
Declaratoria de autenticidad.....	v
Presentación	vi
Índice.....	vii
Índice de Tablas	ix
Índice de Figuras	xiv
Resumen	xxi
Abstract	xxii
I. INTRODUCCIÓN.....	1
1.1. Realidad Problemática	2
1.2.Trabajos Previos.....	7
1.3.Teorías relacionadas al tema.....	15
1.4.Formulación del problema	27
1.5 Justificación de estudio.....	27
1.6 Hipótesis	29
1.7. Objetivos	29
II. MÉTODO	30
2.1. Diseño de la investigación.....	31
2.2.Variables, Operacionalización	33
2.3.Población y Muestra	36
2.4.Técnicas e instrumento de recolección de datos, validez y confiabilidad	38
2.5 Métodos de Análisis de Datos	43
2.6. Aspectos éticos	47
III. RESULTADOS	48

IV. DISCUSIÓN.....	62
V. CONCLUSIONES.....	65
VI. RECOMENDACIONES.....	67
VII. REFERENCIAS	69
VIII. ANEXOS	80
Anexo 1: Matriz de Consistencia	81
Anexo 2: Ficha Técnica instrumentos de recolección de datos PRE-TEST y POS - TEST.....	82
Anexo 3: Fichas de registro.....	83
Anexo 4: Base de Datos Experimental.....	91
Anexo 5: Resultados de la confiabilidad del instrumento	92
Anexo 6: Evaluación de las metodologías para el desarrollo	95
Anexo 7: Entrevista.....	104
Anexo 8: Carta de aprobación de la empresa.....	107
Anexo 9 Metodología OOHDM	108
Fase 1: Obtención de Requerimientos	110
Fase 2: Diseño conceptual	184
Fase 3: Diseño Navegacional.....	188
Fase 4: Diseño de Interfaces Abstractas.	190
Fase 5: Implementación.	202

Índice de Tablas

	Página
Tabla 1 Descripción de criterios para la selección de metodología	21
Tabla 2 Matriz de evaluación de las metodologías por expertos	23
Tabla 3 Comparativa de protocolo de Triage	25
Tabla 4 Operacionalización de las variables	34
Tabla 5 Indicadores	35
Tabla 6 Validación de Instrumento	40
Tabla 7 Índice de correlación de Pearson para indicador Nivel porcentual de pacientes atendidos en triaje.....	42
Tabla 8 Índice de correlación de Pearson para indicador Nivel percentil de cumplimiento.....	42
Tabla 9 Medidas descriptivas del nivel porcentual de pacientes atendidos en traje y después de implementar el Sistema Web utilizando el protocolo de Manchester.....	49
Tabla 10 Medidas descriptivas del nivel percentil de cumplimiento antes y después de la implementación del Sistema Web utilizando el protocolo de Manchester.....	50
Tabla 11 Prueba de Normalidad del nivel porcentual de pacientes atendidos en triaje antes y después de la implementación del Sistema Web utilizando el protocolo de Manchester.....	52
Tabla 12 Prueba de normalidad del nivel percentil de cumplimiento antes y después de implementado el Sistema Web utilizando el protocolo de Manchester.....	54
Tabla 13 Prueba de T-Student para el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica antes y después de implementar el Sistema Web utilizando el protocolo de Manchester.....	57
Tabla 14 Prueba de T-Student para el Nivel percentil de cumplimiento en el proceso de emergencia médica antes y después de implementar el Sistema Web utilizando el protocolo de Manchester	60
Tabla 15 Requerimiento Funcionales	110
Tabla 16 Requerimientos No Funcionales	113
Tabla 17 Relación entre los casos de uso y los requerimientos funcionales.....	114
Tabla 18 Ingresar al Sistema	120
Tabla 19 Registrar Usuario	121

Tabla 20 Actualizar Usuario.....	121
Tabla 21 Registrar Paciente.....	121
Tabla 22 Actualizar Paciente	122
Tabla 23 Crear Orden Atención.....	122
Tabla 24 Listar Orden Atención.....	123
Tabla 25 Ingresar al Sistema	123
Tabla 26 Listar Orden Atención.....	123
Tabla 27 Listar Motivo Atención.....	124
Tabla 28 Registrar Triage	124
Tabla 29 Actualizar Abandono.....	125
Tabla 30 Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage.....	125
Tabla 31 Generar Reporte Triage	125
Tabla 32 Ingresar al Sistema	126
Tabla 33 Listar Orden Atención.....	126
Tabla 34 Registrar Enfermedad Actual	127
Tabla 35 Cargar Enfermedad Actual.....	127
Tabla 36 Actualizar Enfermedad Actual	127
Tabla 37 Registrar Funciones Vitales.....	128
Tabla 38 Cargar Funciones vitales	128
Tabla 39 Actualizar Funciones Vitales	128
Tabla 40 Registrar Diagnóstico	129
Tabla 41 Cargar Diagnóstico	129
Tabla 42 Actualizar Diagnóstico.....	130
Tabla 43 Eliminar Diagnóstico	130
Tabla 44 Registrar Exámenes de Apoyo al Diagnóstico	130
Tabla 45 Cargar Exámenes de Apoyo al Diagnóstico	131
Tabla 46 Actualizar Exámenes de Apoyo al Diagnóstico	131
Tabla 47 Eliminar Exámenes de Apoyo al Diagnóstico	132
Tabla 48 Registrar Receta	132
Tabla 49 Cargar Receta	132
Tabla 50 Actualizar Receta.....	133
Tabla 51 Eliminar Receta	133
Tabla 52 Registrar Descanso Médico	134
Tabla 53 Cargar Descanso Médico	134

Tabla 54 Actualizar Descanso Médico.....	134
Tabla 55 Actualizar Alta Médica.....	135
Tabla 56 Actualizar abandono	135
Tabla 57 Generar Reporte Nivel Percentil de Cumplimiento.....	136
Tabla 58 Generar Reporte Emergencia	136
Tabla 59 Reporte Atención Emergencia.....	136
Tabla 60 Ingresar al Sistema	137
Tabla 61 Listar Orden Atención.....	137
Tabla 62 Registrar Notas Enfermería	138
Tabla 63 Cargar Notas Enfermería.....	138
Tabla 64 Actualizar Notas Enfermería	138
Tabla 65 Actualizar Acto Enfermería	139
Tabla 66 Actualizar abandono	139
Tabla 67 Especificación caso de uso Ingresar al Sistema.....	140
Tabla 68 Especificación caso de uso Gestionar Usuario	141
Tabla 69 Especificación caso de uso Gestionar Paciente	142
Tabla 70 Especificación caso de uso Crear Orden Atención	143
Tabla 71 Especificación caso de uso Listar Orden Atención	145
Tabla 72 Especificación caso de uso Listar Motivo Atención	146
Tabla 73 Especificación caso de uso Registrar Triage.....	147
Tabla 74 Especificación caso de uso Actualizar Abandono	148
Tabla 75 Especificación caso de uso Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage.....	149
Tabla 76 Especificación caso de uso Generar Reporte Triage.....	151
Tabla 77 Especificación caso de uso Gestionar Enfermedad Actual	152
Tabla 78 Especificación caso de uso Gestionar Funciones Vitales	153
Tabla 79 Especificación caso de uso Gestionar Diagnóstico	154
Tabla 80 Especificación caso de uso Gestionar Exámenes de Apoyo al Diagnóstico	156
Tabla 81 Especificación caso de uso Gestionar Receta	157
Tabla 82 Especificación caso de uso Gestionar Descanso Médico	158
Tabla 83 Especificación caso de uso Actualizar Alta Médica	160
Tabla 84 Especificación caso de uso Generar Reporte Nivel Percentil de Cumplimiento	161

Tabla 85 Especificación caso de uso Generar Reporte Emergencia	162
Tabla 86 Especificación caso de uso Actualizar Alta Médica	163
Tabla 87 Especificación caso de uso Gestionar Notas Enfermería	165
Tabla 88 Especificación caso de uso Actualizar Acto Enfermería	166
Tabla 89 Prueba Funcional Ingresar al Sistema	257
Tabla 90 Caso de Prueba Ingresar al Sistema	257
Tabla 91 Prueba Funcional Gestionar Usuario	258
Tabla 92 Caso de Prueba Gestionar Usuario	261
Tabla 93 Prueba Funcional Gestionar Paciente	263
Tabla 94 Caso de Prueba Gestionar Paciente	264
Tabla 95 Prueba Funcional Crear Orden Atención	267
Tabla 96 Caso de Prueba Crear Orden Atención	267
Tabla 97 Prueba Funcional Listar Orden Atención	268
Tabla 98 Caso de Prueba Listar Orden Atención	269
Tabla 99 Prueba Funcional Listar Motivo Atención	270
Tabla 100 Caso de Prueba Listar Motivo Atención	271
Tabla 101 Prueba Funcional Registrar Triage	271
Tabla 102 Caso de Prueba Registrar Triage	274
Tabla 103 Prueba Funcional Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage	276
Tabla 104 Caso de Prueba Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage	276
Tabla 105 Prueba Funcional Generar Reporte Triage	277
Tabla 106 Caso de Prueba Generar Reporte Triage	278
Tabla 107 Prueba Funcional Gestionar Enfermedad Actual	279
Tabla 108 Caso de Prueba Gestionar Enfermedad Actual	279
Tabla 109 Prueba Funcional Gestionar Funciones Vitales	281
Tabla 110 Caso de Prueba Gestionar Funciones Vitales	281
Tabla 111 Prueba Funcional Gestionar Diagnóstico	282
Tabla 112 Caso de Prueba Gestionar Diagnóstico	283
Tabla 113 Prueba Funcional Gestionar Examen de Apoyo al Diagnostico	284
Tabla 114 Caso de Prueba Gestionar Examen de Apoyo al Diagnostico	285
Tabla 115 Prueba Funcional Gestionar Receta	286
Tabla 116 Caso de Prueba Gestionar Receta	287

Tabla 117 Prueba Funcional Gestionar Descanso Médico.....	289
Tabla 118 Caso de Prueba Gestionar Descanso Médico	290
Tabla 119 Prueba Funcional Generar Reporte Nivel Percentil de Cumplimiento	291
Tabla 120 Caso de Prueba Nivel Percentil de Cumplimiento	292
Tabla 121 Prueba Funcional Generar Reporte Emergencia	293
Tabla 122 Caso de Prueba Generar Reporte Emergencia	293
Tabla 123 Prueba Funcional Gestionar Nota Enfermería	294
Tabla 124 Caso de Prueba Gestionar Nota Enfermería	294
Tabla 125 Prueba Funcional Actualizar Abandono.....	295
Tabla 126 Caso de Prueba Actualizar abandono Abandono	296
Tabla 127 Prueba Funcional Actualizar Alta Medica.....	298
Tabla 128 Caso de Prueba Actualizar Alta Medica	298
Tabla 129 Prueba Funcional Reporte Atención Emergencia.....	300
Tabla 130 Caso de Prueba Reporte Atención Emergencia	301
Tabla 131 Prueba Funcional Actualizar Acto Enfermería	302
Tabla 132 Caso de Prueba Acto Enfermería	303

Índice de Figuras

	Página
Figura 1 Causas y Estrategias para mitigar el overcrowding	2
Figura 2 Promedios diarios de atención en la Emergencia de Adultos Hospital Nacional Rebagliati, EsSalud	3
Figura 3 Nivel porcentual de pacientes atendidos en triaje	6
Figura 4 Nivel percentil de cumplimiento	7
Figura 5 Proceso de Emergencia Medica	16
Figura 6 Nivel porcentual de pacientes atendidos en triaje	18
Figura 7 Formula Nivel percentil de cumplimiento indicador 2	19
Figura 8 Arquitectura Web	20
Figura 9 Niveles de Clasificación Manchester	26
Figura 10 Diseño de investigación	32
Figura 11 Grado de confiabilidad	41
Figura 12 Porcentaje del nivel porcentual de pacientes atendidos en triaje antes y después de implementar el Sistema Web utilizando el protocolo de Manchester	50
Figura 13 Porcentaje del nivel percentil de cumplimiento antes y después de la implementar el Sistema Web utilizando el protocolo de Manchester	51
Figura 14 Prueba de Normalidad del nivel porcentual de pacientes atendidos en triaje antes de implementar el Sistema Web utilizando el protocolo de Manchester	53
Figura 15 Prueba de normalidad del nivel porcentual de pacientes atendidos en triaje de después de implementar el Sistema Web utilizando el protocolo de Manchester	53
Figura 16 Prueba de normalidad del nivel percentil de cumplimiento antes de implementar el Sistema Web utilizando el protocolo de Manchester	55
Figura 17 Prueba de normalidad del nivel percentil de cumplimiento después de implementar el Sistema Web utilizando el protocolo de Manchester	55
Figura 18 Nivel Porcentual de Pacientes Atendidos en Triage - Comparativa General	57
Figura 19 Prueba T-Student – Nivel porcentual de pacientes atendidos en triaje	58
Figura 20 Nivel percentil de cumplimiento - Comparativa General	59

Figura 21 Prueba T-Student – Nivel percentil de cumplimiento	61
Figura 22 Caso de uso admisionista	116
Figura 23 Caso de uso internista	117
Figura 24 Caso de uso medico	117
Figura 25 Caso de uso enfermera	118
Figura 26 Diagrama de Secuencia Ingresar al Sistema.....	167
Figura 27 Diagrama de Secuencia Actualizar Usuario	168
Figura 28 Diagrama de Secuencia Registrar Usuario.....	168
Figura 29 Diagrama de Secuencia Actualizar Paciente	169
Figura 30 Diagrama de Secuencia Registrar Paciente	169
Figura 31 Diagrama de Secuencia Listar Orden Atención - Admisionista	170
Figura 32 Diagrama de Secuencia Crear Orden Atención	170
Figura 33 Diagrama de Secuencia Listar Orden Atención - Médico.....	171
Figura 34 Diagrama de Secuencia Listar Orden Atención - Internista.....	171
Figura 35 Diagrama de Secuencia Listar Motivo Atención	172
Figura 36 Diagrama de Secuencia Listar Orden Atención – Enfermera	172
Figura 37 Diagrama de Secuencia Registrar Triage.....	173
Figura 38 Diagrama de Secuencia Abandono – Internista	174
Figura 39 Diagrama de Secuencia Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage.....	175
Figura 40 Diagrama de Secuencia Generar Reporte Triage	175
Figura 41 Diagrama de Secuencia Gestionar Funciones Vitales	176
Figura 42 Diagrama de Secuencia Gestionar Enfermedad Actual	176
Figura 43 Diagrama de Secuencia Gestionar Diagnóstico	177
Figura 44 Diagrama de Secuencia Gestionar Exámenes de Apoyo al Diagnostico	178
Figura 45 Diagrama de Secuencia Gestionar Receta	179
Figura 46 Diagrama de Secuencia Abandono - Médico	180
Figura 47 Diagrama de Secuencia Gestionar Descanso Médico	180
Figura 48 Diagrama de Secuencia Generar Reporte Atención.....	181
Figura 49 Diagrama de Secuencia Alta Medica	181
Figura 50 Generar Reporte Nivel Percentil de Cumplimiento	182
Figura 51 Diagrama de Secuencia Generar Reporte Emergencia	182
Figura 52 Diagrama de Secuencia Acto Enfermería.....	183

Figura 53 Diagrama de Secuencia Gestionar Nota Enfermería	183
Figura 54 Secuencia Abandono - Enfermería	184
Figura 55 Modelo Entidad Relación.....	185
Figura 56 Diagrama lógico de base de datos	186
Figura 57 Diagrama físico de base de datos	187
Figura 58 Diagrama navegacional de admisionista	188
Figura 59 Diagrama navegacional de internista	189
Figura 60 Diagrama navegacional de medico	189
Figura 61 Diagrama navegacional de enfermera	190
Figura 62 Diseño de Interfaz Ingresar al Sistema.....	191
Figura 63 Diseño de Interfaz Gestionar Usuario	191
Figura 64 Diseño Interfaz Gestionar Pacientes	192
Figura 65 Diseño Interfaz Crear Orden Atención	192
Figura 66 Diseño Interfaz Listar Ordenes de Atención en Bandeja Admisionista.....	193
Figura 67 Diseño Interfaz Listar Ordenes de Atención en Bandeja Internista.....	193
Figura 68 Diseño Interfaz Listar Ordenes de Atención en Bandeja Medico.....	194
Figura 69 Diseño Interfaz Listar Ordenes de Atención en Bandeja Enfermera	194
Figura 70 Diseño de Interfaz Motivos Atención.....	195
Figura 71 Diseño de Interfaz Registrar Triage.....	195
Figura 72 Diseño de Interfaz Reporte Nivel Porcentual Pacientes Atendidos en Triage	196
Figura 73 Diseño de Interfaz Reporte Triage	196
Figura 74 Diseño de Interfaz Gestionar Enfermedad Actual	197
Figura 75 Diseño Interfaz Gestionar Funciones Vitales	197
Figura 76 Diseño de Interfaz Gestionar Diagnostico	198
Figura 77 Diseño Interfaz Gestionar Examen de Apoyo al Diagnostico	198
Figura 78 Diseño de Interfaz Gestionar Receta	199
Figura 79 Diseño de Interfaz Gestionar Descanso Medico	199
Figura 80 Diseño de Interfaz Reporte Nivel Percentil de Cumplimiento	200
Figura 81 Diseño de Interfaz Alta Médica	200
Figura 82 Diseño de Interfaz Reporte Emergencia.....	201
Figura 83 Diseño de Interfaz Gestionar Notas de Enfermería	201
Figura 84 Modelo Usuario	202

Figura 85 Vista Ingresar al Sistema	203
Figura 86 Controlador Ingresar al Sistema	203
Figura 87 Ejecución BD Ingresar al Sistema	204
Figura 88 Modelo Persona	204
Figura 89 Vista Gestionar Usuario	205
Figura 90 Controlador Gestionar Usuario	205
Figura 91 Ejecución BD Registrar Usuario	206
Figura 92 Modelo Paciente	206
Figura 93 Vista Gestionar Paciente	207
Figura 94 Controlador Gestionar Paciente	207
Figura 95 Ejecución BD Registrar Paciente	208
Figura 96 Modelo Orden Atención	208
Figura 97 Vista Crear Orden Atención	209
Figura 98 Controlador Crear Orden Atención	209
Figura 99 Ejecución BD Crear Orden Atención	210
Figura 100 Vista Listar Orden Atención	210
Figura 101 Controlador Listar Orden Atención	211
Figura 102 Ejecución BD Listar Orden Atención	211
Figura 103 Modelo Prioridad	212
Figura 104 Vista Motivo Atención	213
Figura 105 Vista Listar Motivo Atención	213
Figura 106 Vista Listar Motivo Atención	214
Figura 107 Controlador Listar Motivo Atención	214
Figura 108 Ejecución BD Listar Motivo Atención	215
Figura 109 Modelo Triage	216
Figura 110 Vista Registrar Triage	216
Figura 111 Controlador Triage	217
Figura 112 Ejecución BD Registrar Triage	217
Figura 113 Ejecución Registrar Triage	218
Figura 114 Vista Abandono	218
Figura 115 Controlador Abandono	219
Figura 116 Ejecución BD Abandono	219
Figura 117 Modelo Nivel Porcentual de Pacientes Atendidos en Triage	220
Figura 118 Vista Nivel Porcentual de Pacientes Atendido en Triage	220

Figura 119 Controlador Nivel Porcentual de Pacientes Atendidos en Triage	221
Figura 120 Ejecución BD Nivel Porcentual Pacientes Atendidos en Triage.....	221
Figura 121 Modelo Reporte Triage	222
Figura 122 Vista Reporte Triage	222
Figura 123 Controlador Reporte Triage	223
Figura 124 Ejecución BD Reporte Triage	223
Figura 125 Modelo Enfermedad Actual	224
Figura 126 Vista Enfermedad Actual.....	224
Figura 127 Controlador Enfermedad Actual	225
Figura 128 Ejecución BD Registrar Enfermedad Actual.....	225
Figura 129 Modelo Funciones Vitales	226
Figura 130 Vista Funciones Vitales	226
Figura 131 Controlador Funciones Vitales	227
Figura 132 Ejecución BD Registrar Funciones Vitales.....	227
Figura 133 Modelo Diagnostico	228
Figura 134 Vista Diagnóstico	228
Figura 135 Controlador Diagnóstico	229
Figura 136 Ejecución BD Registrar Diagnóstico.....	229
Figura 137 Modelo Examen de Apoyo al Diagnostico	230
Figura 138 Vista Examen de Apoyo al Diagnostico	230
Figura 139 Controlador Examen de Apoyo al Diagnostico	231
Figura 140 Ejecución Registrar Examen de Apoyo al Diagnóstico	231
Figura 141 Modelo Receta	232
Figura 142 Vista Receta	232
Figura 143 Controlador Receta	233
Figura 144 Ejecución BD Registrar Receta.....	233
Figura 145 Modelo Descanso Medico	234
Figura 146 Vista Descanso Medico	234
Figura 147 Controlador Descanso Medico	235
Figura 148 Ejecución BD Registra Descanso Medico	235
Figura 149 Modelo Alta Medica	236
Figura 150 Vista Alta Medica	236
Figura 151 Controlador Alta Medica	237
Figura 152 Ejecución BD Alta Medica	237

Figura 153 Modelo Nivel Percentil de Cumplimiento	238
Figura 154 Vista Nivel Percentil de Cumplimiento.....	238
Figura 155 Controlador Nivel Percentil de Cumplimiento	239
Figura 156 Ejecución BD Nivel Percentil de Cumplimiento	239
Figura 157 Modelo Reporte Emergencia.....	240
Figura 158 Vista Reporte Emergencia	240
Figura 159 Controlador Reporte Emergencia	241
Figura 160 Ejecución BD Reporte Emergencia	241
Figura 161 Modelo Nota Enfermería	242
Figura 162 Vista Nota Enfermería	242
Figura 163 Controlador Nota Enfermería	243
Figura 164 Ejecución BD Nota Enfermería	243
Figura 165 Vista Acto Enfermería	244
Figura 166 Controlador Acto Enfermería.....	244
Figura 167 Ejecución BD Acto Enfermería.....	245
Figura 168 Interfaz de Ingresar al Sistema	245
Figura 169 Interfaz Gestionar Usuario	246
Figura 170 Interfaz Gestionar Paciente	246
Figura 171 Interfaz Crear Orden Atención	247
Figura 172 Interfaz Listar Orden Atención en Bandeja Admisionista	247
Figura 173 Interfaz Listar Orden Atención en Bandeja Internista	248
Figura 174 Interfaz Listar Orden Atención en Bandeja Médico	248
Figura 175 Interfaz Listar Orden Atención en Bandeja Enfermera	249
Figura 176 Interfaz Motivos de Atención.....	249
Figura 177 Interfaz Registrar Triage.....	250
Figura 178 Interfaz Reporte Nivel Porcentual de Pacientes Atendidos en Triage	250
Figura 179 Interfaz Reporte Triage	251
Figura 180 Interfaz Enfermedad Actual	251
Figura 181 Interfaz Funciones Vitales.....	252
Figura 182 Interfaz Diagnóstico	252
Figura 183 Interfaz Exámenes de Apoyo al Diagnóstico	253
Figura 184 Interfaz Receta	253
Figura 185 Interfaz descanso médico	254

Figura 186 Interfaz Reporte Nivel Percentil de Cumplimiento	254
Figura 187 Interfaz Reporte Emergencia.....	255
Figura 188 Interfaz alta medica	255
Figura 189 Reporte de atención emergencia	256
Figura 190 Interfaz Notas Enfermería	256

RESUMEN

La presente tesis detalla el desarrollo del Sistema Web utilizando el Protocolo de Manchester para el proceso de Emergencia Médica de la Clínica Sanna Borja, debido a que la situación de la Clínica previa a la aplicación del sistema presentaba deficiencias en cuanto al nivel porcentual de pacientes atendidos en triaje y el nivel percentil de cumplimiento. El objetivo de esta de esta investigación fue Determinar la influencia del sistema web utilizando el protocolo de Manchester en el proceso de emergencia médica en la Clínica Sanna San Borja.

Por ello, se describe previamente aspectos teóricos de lo que es el proceso de emergencia médica. Así como las metodologías que se utilizaron para el desarrollo del sistema web. Para el desarrollo del sistema web, se empleó la metodología OOHDM, por ser la que más se adapta a las necesidades y etapas del proyecto, además, se utilizó el lenguaje de programación Java con el Spring framework 3.9.2 y el gestor de base de datos SQL Server 2012 R2.

El tipo de investigación es aplicada - experimental, tipo de estudio es explicativo, tipo diseño de la investigación es pre- experimental y el enfoque es cuantitativo. La técnica de recolección de datos fue el fichaje y el instrumento fue la ficha de registro, los cuales fueron validados por expertos. La población para el nivel porcentual de pacientes atendidos en triaje se determinó en 1493 registros de triaje agrupados en 20 reportes. El tamaño de la muestra estuvo conformado por 306 registros de triaje agrupados en 20 reportes de un mes estratificados por día. La población para el percentil de cumplimiento se determinó en 1211 registros de triaje agrupados en 20 reportes. El tamaño de la muestra estuvo conformado por 292 registros de triaje agrupados en 20 reportes de un mes estratificados por día. El muestreo es aleatorio probabilístico simple, los cuales fueron validados por un juicio de expertos.

La implementación del Sistema web utilizando el protocolo de Manchester permitió incrementar el nivel porcentual de pacientes atendidos en triaje de un 69.12% al 93.81%, del mismo modo el nivel percentil de cumplimiento del 65.88% al 89.47%. Los resultados mencionados anteriormente, permitieron llegar a la conclusión que el Sistema web utilizando el protocolo de Manchester mejora el proceso de emergencia médica de la Clínica Sanna San Borja.

Palabras clave: emergencia médica, protocolo de Manchester, metodología OOHDM, percentil de cumplimiento y nivel porcentual de pacientes atendidos en triaje.

ABSTRACT

This thesis details the development of the Web System using the Manchester Protocol for the Medical Emergency process of the Sanna Borja Clinic, because the situation of the Clinic prior to the application of the system presented deficiencies in terms of the percentage level of patients treated in triage and the percentile level of compliance. The objective of this research was to determine the influence of the web system using the Manchester protocol in the medical emergency process at the Sanna San Borja Clinic.

Therefore, theoretical aspects of what is the medical emergency process is described previously. As well as the methodologies that were used for the development of the web system. For the development of the web system, the OOHDM methodology was used, because it is the one that best suits the needs and stages of the project, in addition, the Java programming language was used with the Spring framework 3.9.2 and the base manager of SQL Server 2012 R2 data.

The type of research is applied - experimental, type of study is explanatory, type of research design is pre-experimental and the approach is quantitative. The data collection technique was the signing and the instrument was the registration form, which were validated by experts. The population for the percentage level of patients treated in triage was determined in 1493 triage records grouped into 20 reports. The sample size consisted of 306 triage records grouped into 20 one-month reports stratified per day. The population for the compliance percentile was determined in 1211 triage records grouped into 20 reports. The sample size consisted of 292 triage records grouped into 20 one-month reports stratified per day. The sampling is simple random probabilistic, which were validated by an expert judgment.

The implementation of the Web System using the Manchester protocol allowed to increase the percentage level of patients treated in triage from 69.12% to 93.81%, in the same way the percentile level of compliance from 65.88% to 89.47%. The results mentioned above, allowed us to conclude that the Web System using the Manchester protocol improves the medical emergency process of the Sanna San Borja Clinic.

Keywords: medical emergency, Manchester protocol, OOHDM methodology, compliance percentile and percentage level of patients treated in triage.

I. INTRODUCCIÓN

1.1. Realidad Problemática

Según Tobar, Retamal y Garrido (2014) Los servicios de emergencia son partes claves dentro del desarrollo hospitalario. Es un servicio con demanda voluble donde se realiza la atención de los pacientes las 24 horas del día con diferente grado de diagnóstico y exigencias, manifestando una gran fracción de las admisiones. En los últimos 5 años, y por motivo a las diversas variaciones epidemiológicas de la sociedad, a la variación de los requerimientos y apreciaciones de la sociedad, como también a la falla en los procedimientos de atención médica, se ha experimentado una gran demanda en las atenciones en estos servicios, lo que ha conducido al fenómeno conocido como overcrowding (saturación o colapso) en estos servicios. Esto se justifica por diversos motivos (Figura 1), recalcando entre ellas la poca capacidad de resolución por los sistemas de Atención Primaria y a los cambios en la instrucción de los pacientes a ser clientes en salud. Actualmente, y en acuerdo con la “Ley de Deberes y Derechos de los Pacientes”, la expectativa de los pacientes y familias no es sólo saber su diagnóstico y su terapia adecuada, sino que además la atención sea conveniente y accesible. Estos dos factores mencionados son muy sobresalientes en el SE, donde a menudo hay momentos de demora y presión emocional de pacientes desvalidos. La tiempo-dependencia de los cuidados en los SE es probablemente una de las particularidades esenciales de estos servicios. (p.189-190)

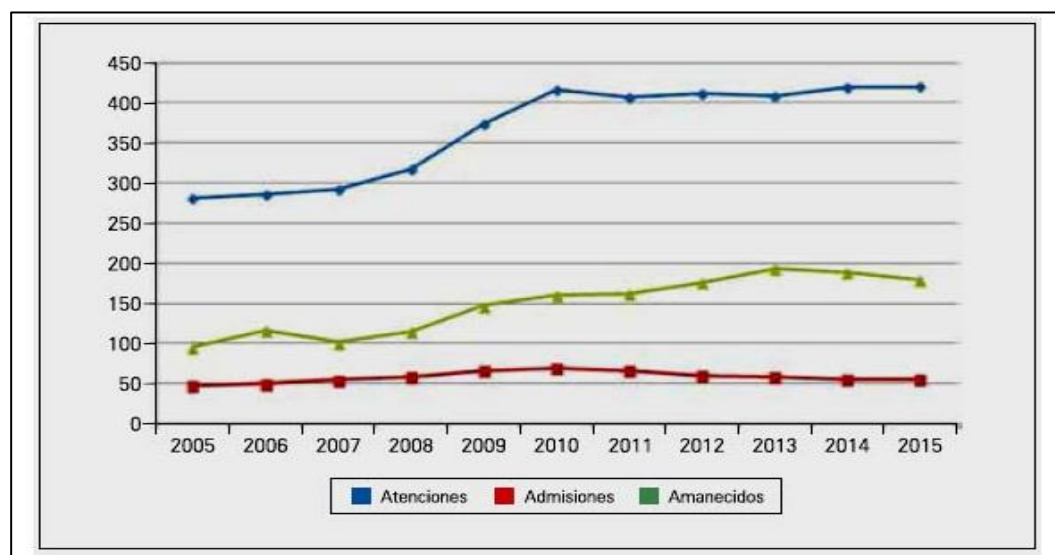
Figura 1 Causas y Estrategias para mitigar el overcrowding

Causa	Estrategia para mitigar
Tiempos elevados de hospitalización	<ul style="list-style-type: none">• Implementar una sala de pre hospitalización, donde esperen los pacientes hasta que se libere una cama.• Aumentar la capacidad de camas de hospitalización o reservar camas exclusivas para urgencia.• Mejorar la gestión de camas del hospital para aumentar su disponibilidad.
Exceso de exámenes y elevados tiempos de respuesta	<ul style="list-style-type: none">• Creación de protocolos que definan claramente en qué situaciones se debe pedir un examen de apoyo diagnóstico.• Controlar y monitorear de manera colaborativa con las unidades de apoyo los tiempos de respuesta de los exámenes, definiendo metas y acciones de mejora.
Capacidad insuficiente de recursos: RRHH e infraestructura	<ul style="list-style-type: none">• Generar proyectos sustentables que busquen aumentar la capacidad del servicio.• Reasignar de mejor forma los recursos actuales, adaptándolos según la distribución de la demanda.
Problemas de comunicación y coordinación	<ul style="list-style-type: none">• Implementar un sistema de monitoreo en tiempo real que permita conocer el estado de las actividades oportunamente.• Diseñar los procesos de atención, definiendo claramente los responsable de cada actividad.
Mecanismos de control insuficientes	<ul style="list-style-type: none">• Definir una serie de indicadores de diferentes perspectivas que permitan medir el desempeño del servicio (Ejemplo: balance scorecard).• Realizar un seguimiento a los indicadores y fijar metas factibles para ir mejorando.

Fuente: Revista Hospital Clínico Universidad de Chile, 2014.

Según Vásquez et al. (2016) La unidad de emergencia actualmente es de suma importancia en el enfoque del régimen de salud. La obligación de tener una rápida atención, en percibir oportunamente los diagnósticos y terapias, es uno de los elementos por los que el manejo del servicio de emergencia ha incrementado actualmente. A su vez estas prestaciones se han ido acoplando a este aumento de la demanda, descubriendo diversas formas de cuidado médico, grupos de modelos asistenciales y uso de la ciencia destacada. Pues se creó un problema demandante reproducido en el mundo, denominado apilamiento (overcrowding) de las unidades de Emergencia. Esto se debe al uso inapropiado de las Emergencias, pues conlleva a un aumento de las atenciones del servicio de emergencia. El paciente al ingresar se le determina una prioridad (área de triaje) y se produce una historia de emergencia con datos normalizados de nombre, edad, sexo y fecha de ingreso. Después, el paciente es trasladado al área seleccionada, donde el médico inspecciona la historia clínica y registra en forma manual la atención, además indica el traslado del paciente, después la historia clínica retorna al archivero, este registra las evaluaciones de la atención del médico y en el sistema estadístico. Evaluando el reporte del sistema estadístico, se verificó los reportes hasta el año 2015 de pacientes registrados en el box de observaciones (Figura 02). (p. 379-380).

Figura 2 Promedios diarios de atención en la Emergencia de Adultos Hospital Nacional Rebagliati, EsSalud



Fuente: Vásquez et al. (2016)

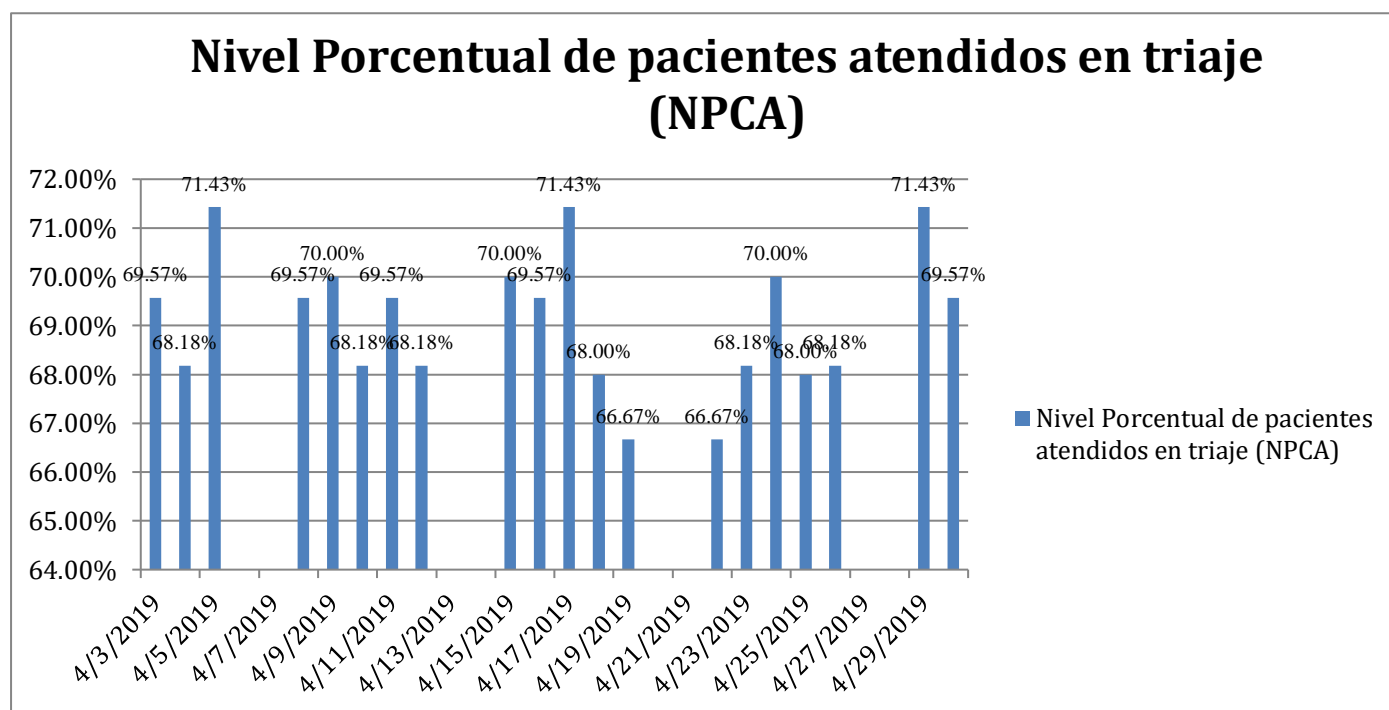
La Clínica Sanna San Borja, ubicada en Av. Guardia Civil 337 es una entidad con una infraestructura amplia y de calidad de servicio para la salud del paciente y buscar siempre

su bienestar. La visión fundamental de la clínica es ser una de las entidades de salud más conocidas del país obteniendo los estándares de calidad y seguridad médica. El área de emergencia cuenta con un servicio implementado con todos los equipos necesarios para una atención oportuna, cuenta con una unidad de Trauma - shock, tópico para cirugía menor – Poli traumatizados, así también atención médica en Pediatría, medicina interna, cirugía y Cardiología, etc. Para obtener la información preliminar se aplicó el levantamiento de información a través de la entrevista que se le realizó al Ing. Charlie Ramírez (anexo 7) y así obtener las observaciones correspondientes. El proceso médico de emergencia inicia cuando el paciente ingresa debido a algún motivo por el cual debe ser atendido, lo primero a realizar es la admisión de la atención para lo cual el paciente o acompañante debe entregar el documento de identificación para así poder ingresar los datos necesarios para crear la orden de atención, en algunos casos el paciente no lleva consigo el documento que lo pueda identificar para estos casos se toma en consideración otro tipo de documento que lo identifique o se anota los datos necesarios es una ficha, pero este es un problema ya que genera demora para poder registrarlo por sistema ya que el DNI es una información necesaria para poder registrarlo en el sistema actual que maneja la clínica, este es un sistema de escritorio el cual no está programado para poder realizarle cambios. Luego de haber registrado al paciente ya sea por el sistema o la ficha este es derivado al área de triaje, si el paciente ingresa por un motivo muy grave se deriva directo al tópico de trauma shock y el registro de los datos de la atención se ingresan después de haber atendido este caso. Al tener ya la orden de atención como ya se mencionó el paciente debe dirigirse al área de triaje en el cual se evaluará sus signos vitales, se ingresa el motivo de ingreso, antecedentes, alergias, box correspondiente para la atención de emergencia y se asigna la prioridad del 1 al 4, pero no siempre el registro es el correcto, debido a la congestión (overcrowding) de pacientes esto genera una confusión en su registro en el internista a cargo de la clasificación de triaje. Actualmente los datos de la atención del triaje se ingresan de forma manual en una ficha y se adjunta en el historial de la atención, además se le coloca al paciente una cinta de color de acuerdo a la prioridad asignada, si tuviese alergia se le coloca una cinta roja adicional. Además, en este módulo las fichas de registro de triaje pueden ser extraviadas y esto surge problemas ya en la atención de la emergencia, debido a que genera demora en el tiempo para poder atender al paciente. Después de pasar por triaje la enfermera guía al paciente al box correspondiente a su motivo de ingreso ya sea: pediátrico, adultos, traumatología,

otros. En el box asignado debe atender el médico de turno o especialista. Los internistas ingresan al sistema de escritorio, ingresan la asignación de la prioridad y el box que le corresponde al paciente para que luego el médico de turno o especialista pueda visualizar en la pantalla a los pacientes que tienen en cola, otro problema en esta fase es que los internistas pueden ingresar datos erróneos y no se lleva un control de los boxes ocupados en el momento por parte del sistema actual. Otra observación que se encontró es que al ingresar mal un dato fundamental como es la prioridad puede llevar al error de lograr una atención rápida y aquí es donde se genera la mayor molestia por parte del paciente, ya que en algunos casos deben esperar a ser atendidos más del tiempo que les corresponde, además en la pantalla que visualiza el médico observa que sale ordenado de acuerdo al tiempo de ingreso y no a la prioridad de atención, este cambio no se puede realizar, porque el procedimiento actual no lo aprueba. Los médicos por su parte no se guían mucho por el orden mostrado en la pantalla; sino de la prioridad, pero sería de mucha ayuda si es que se pudiese ordenar de acuerdo a la prioridad. El médico de turno o especialista se asigna a la atención y solo él puede realizar la atención a no ser que realice una interconsulta. Al ingresar a la atención el médico registra en los formatos que posee el sistema los cuales son: Anamnesis, antecedentes, alergias, diagnóstico, receta, exámenes, descanso médico e interconsulta. Estos formatos no están completos para el médico, además indica que el sistema es muy tedioso y de difícil acceso para realizar la atención en algunos casos opta por registrar la atención del paciente de forma manual en sus fichas correspondientes. En el caso de la enfermera, también realiza su propio registro en unas fichas nombradas como notas de enfermería en las cuales indica la evolución del paciente como si fuese un historial, además añade algún medicamento en la ficha si fuese el caso. Al realizar el registro de la atención por parte de la enfermera, ya el médico puede realizar el alta del paciente y brinda una receta de alta o un descanso médico si es que fuese el caso. Actualmente estas fichas se registran de forma manual ya que no existe un módulo en el sistema de escritorio para que la enfermera pueda registrar su atención, además esto es una dificultad, ya que existen casos que las fichas puedan extraviarse y es importante que el médico tenga un módulo en el cual pueda observar las anotaciones de la enfermera de forma eficaz y confiable sobre la evolución del paciente mientras sigue en emergencia. Por otra parte, existen casos en los cuales los pacientes abandonan la atención ya sea en el módulo de triaje o en la emergencia, el sistema actual no posee un estado que indique que el paciente ha abandonado la atención, actualmente solo se indica de forma manual

en las fichas de registro que el paciente ha abandonado la atención o en el sistema los médicos registran en el campo de relato del formulario de anamnesis que el paciente ha abandonado la atención de forma voluntaria. Asimismo, uno de los problemas encontrados tiene que ver con el nivel porcentual de pacientes atendidos en triaje, desde que son admisionados hasta que se realiza la priorización del triaje, aquí hay un leve tiempo de espera de los pacientes el cual es un porcentaje alto, por el cual se ha identificado un cuello de botella en este intervalo, actualmente se verifica que se encuentra en un 69% el total de porcentaje este valor debería superar en un 85% de acuerdo al estándar del SET (sistema estructurado de triaje) evidenciado en la figura 3.

Figura 3 Nivel porcentual de pacientes atendidos en triaje

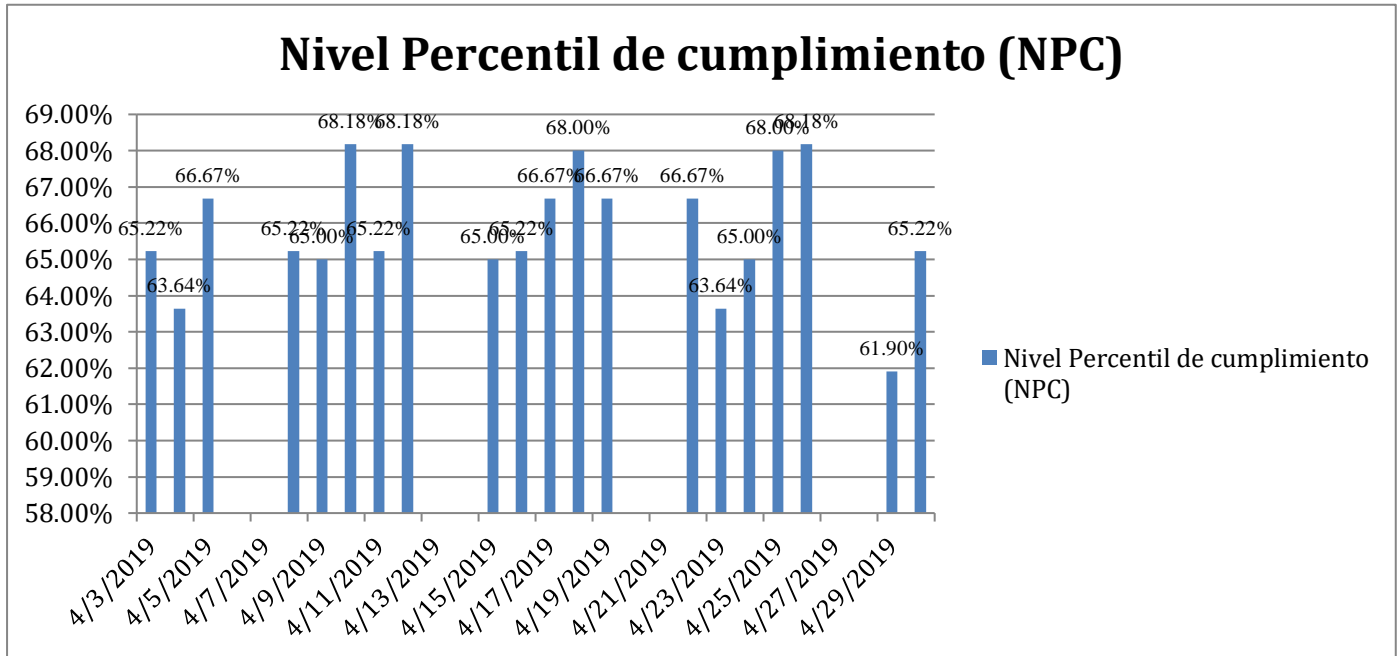


Fuente: Elaboración Clínica Sanna San Borja (2019)

Otro problema asociado es el nivel percentil de cumplimiento, es la etapa en donde el paciente realiza la espera del médico de emergencia después de haberle sido asignado una prioridad en el área de triaje. Estos tiempos de espera dependen de la prioridad asignada, pero en muchos casos no se cumplen con el tiempo establecido y en porcentaje no cumplen con el nivel percentil de cada prioridad y esto genera la molestia del paciente, ya que causa su incomodidad y molestia al no saber qué es lo tiene, actualmente el nivel

percentil de cumplimiento está en un 66% evidenciado en la figura 4, de acuerdo a estándar el mínimo porcentaje debe ser un 70% del nivel percentil de cumplimiento.

Figura 4 Nivel percentil de cumplimiento



Fuente: Elaboración Clínica Sanna San Borja (2019)

¿Qué sucederá si se sigue teniendo los mismos problemas la Clínica Sanna San Borja?, en respuesta a dicha pregunta, generaría un colapso en las atenciones por el tiempo en la demora por consiguiente los pacientes no serán atendidos a tiempo y los médicos no podrán atender, al suscitarse ello realizarán reclamos y posterior a eso dejarán de ir a la clínica, así mismo la clínica pagará las multas generada por los reclamos, por lo tanto, se generará una mala perspectiva de la institución.

1.2.Trabajos Previos

Internacionales

Según Muñoz et al. (2015), en el artículo “Sistema informático para el manejo de la información hospitalaria sobre infarto agudo de miocardio”, en la Universidad de Ciencias Médicas de Cienfuegos, Cienfuegos-Cuba. El alcance del estudio es desarrollar un programa en entorno web que permita el manejo de la información sobre el diagnóstico infarto agudo de miocardio en la especialidad de cardiología para hacerse con la información y tener una estadística de los datos registrados de la atención del paciente, debido a que la terapia del infarto agudo de miocardio se encuentra en constante cambio

esta información se obtiene de los diversos procedimientos y pruebas de desarrollo por cardiólogos. El desarrollo de la metodología utilizada fue la de scrum, ya que es una metodología iterativa que vale para gestionar y administrar el progreso de un sistema. El modelado fue desarrollado en UML para simbolizar de los casos de uso y del sistema, además que sirve para documentar el proyecto. Como muestra se obtiene los registros del Hospital Gustavo Aldereguia Lima de Cienfuegos. El presente estudio utilizó un diseño experimental-aplicado y una investigación cuantitativa. La población se determinó en 1300 y con una muestra de 297 de los registros de pacientes atendidos. Para conseguir los datos de las variables, se realizó las técnicas de entrevista, encuesta y análisis. Entre los primordiales resultados conseguidos se manifiesta que se requiere un Sistema informático para el manejo de la información hospitalaria. Precedentemente de la implementación del sistema informático un 78% cumplían las atenciones a los pacientes atendidos, y posteriormente de la implementación del sistema informático un 89%, hubo un incremento del 11% de las atenciones a los pacientes atendidos, después de un análisis clínico y epidemiológico del IAM se definió para el registro aquellas que se relacionen con las guías prácticas: Datos generales del paciente y antecedentes patológicos personales. Esta investigación concluye con el desarrollo del sistema que permite controlar la data sobre el diagnóstico infarto agudo de miocardio y sus atenciones. El desarrollo es compatible al sistema cubano el cual posee las propiedades para el control de la salud y la demografía de la sociedad de Cuba.

Esta investigación sirvió para contribuir en la importancia de implementar un sistema informático que posibilite gestionar la información de la atención de un paciente, además nos indica el valor porcentual de una atención y como el sistema informático puede mejorar ese equivalente.

Torque y Rodrigues (2013), realizó la tesis denominada “desenvolvimento de um Sistema de Informação para Triagem com Classificação de risco nos Serviços de Urgência e Emergência”, desarrollada en la Universidade do sul de Santa Catarina, en Palhoça - Brasil, la investigación tiene como objetivo sistematizar el proceso de triaje en el Hospital Odilon Behrens utilizando el protocolo de Manchester, establecer niveles de prioridad para los pacientes en servicios de urgencia y de emergencia del Sistema Único de Salud (SUS), dentro de la investigación se puede señalar que todavía hay problemas con respecto a la manera de cómo se recibe al paciente en los servicios de salud, además existe

una ineficiencia en la clasificación en triage que pone en riesgo la salud del paciente, la falta de infraestructura, tanto física como de personal, complica aún más el proceso de la atención en las unidades del proceso emergencia médica, éstos los cuales son puertas de entrada al sistema de salud, entre estos protocolos de evaluación de riesgos existentes, el Protocolo Manchester es uno de los más completos para dar solución a este problema. Se utilizó un diseño pre experimental tipo aplicada con un enfoque cuantitativo. Se realizó para esta investigación las técnicas de entrevista, encuesta y análisis, La población utilizada de pacientes atendidos es de 980 y con una muestra de 276. Entre los primordiales resultados conseguidos se manifiesta que se requiere sistema de información basado en el triaje. Precedentemente de la implementación del sistema informático un 68% cumplían el servicio de las atenciones a los pacientes en emergencia, y posteriormente de la implementación del sistema informático un 90%, hubo un incremento del 22% del cumplimiento de atenciones de los pacientes en emergencia. La investigación concluye en implementar un SI que se basa en la clasificación del protocolo de Manchester para el cuidado de salud en unidades de emergencia y de emergencia.

Este antecedente nos da un aporte sobre el resultado de la utilización del protocolo de Manchester dentro del proceso de emergencia, también nos muestra que en otros países el servicio de emergencia médica puede ser considerado de complejidad en el cuidado de salud debido a la alta demanda que existe.

Monroy y Silva (2018), en la tesis cuyo título es “Desarrollo e implementación de un sistema web para el Hospital León Becerra de Guayaquil enfocado en el registro de emergencia utilizando el lenguaje de programación php y el framework laravel”, en la Universidad de Guayaquil, Ecuador. El objetivo de la investigación es desarrollar un sistema en ambiente web mediante el framework laravel que permite registrar las atenciones de emergencia que se llevan a cabo en el Hospital León Becerra de Guayaquil, debido a que actualmente tienen un sistema de escritorio que siempre debe actualizarse y el cual demora aproximadamente 10 minutos, tomando en cuenta que otras deben esperar que culmine la jornada laboral para iniciar la actualización, debido a que los usuarios acceden al sistema y no deben dejar de lado sus otras actividades, ya que estas actualizaciones se ejecutan mínimo una vez a la semana por la pérdida de tiempo que ocasionan y el costo monetario que significa para el hospital. Asimismo, el problema general es que debido a estas actualizaciones el personal asistencial no puede proseguir

con sus actividades en su sistema actual; por lo que, deben optar por atender de una manera tradicional, debido a este problema en ocasiones genera perdida de información y sobre todo perdida de su tiempo. Se utilizó un diseño pre experimental tipo aplicada con un enfoque cuantitativo. Se realizó para esta investigación las técnicas de entrevista, encuesta y análisis. La población utilizada de pacientes atendidos es de 370 y con una muestra de 188. Precedentemente de la implementación del sistema web es un 62% cumplían el servicio de las atenciones a los pacientes en emergencia, y posteriormente de la implementación del sistema web un 80%, hubo un aumento del 18% del cumplimiento de atenciones de los pacientes en emergencia. La investigación concluye que, al desarrollar un sistema web, se cubrió las necesidades en la atención de emergencia y además mejoro el acceso del usuario con el sistema y sus atenciones.

Esta investigación brinda un gran aporte, ya que evidencia que la utilización de un sistema en entorno web es más amigable para el usuario y mejora los procesos de la atención en emergencia.

Alvarenga Molina et al. (2015), realizo la tesis denominada “Sistema de Información para el área de Hospitalización en el Hospital Nacional Rosales”, desarrollada en la Universidad de El Salvador, en San Salvador, esta investigación tiene como objetivo implementar un SI para el servicio de Hospitalización en el Hospital Nacional Rosales que almacene la información de los procesos y tareas diarias que se efectúan en este servicio, debido a que existen dificultades de los mandos superiores del HNR para conseguir información del servicio de Hospitalización para la toma de decisiones, al tener todo un registro en torno manual existen perdidas de la información correspondiente a las atenciones a los pacientes. La información obtenida posee una demora de 1 a 2 meses para ser reforzada y exhibida, debido a que el cuerpo médico encomendado para esta tarea, invierte demasiadas horas en esta actividad, en donde carga una cantidad de datos voluminosa, originando que la data no se obtenga en el tiempo esperado, aplazando el envío de los reportes habituales para las jefaturas del hospital sobre el desarrollo de los procesos del cuerpo médico ejecutados en el servicio de Hospitalización. Se realizó para esta investigación las técnicas de entrevista, encuesta y análisis. La población utilizada de pacientes atendidos es de 420 y con una muestra de 201. Los resultados precedentemente del desarrollo del sistema web es de 64% cumplían el servicio de las atenciones a los pacientes en emergencia, y posteriormente de la implementación del

sistema web un 81%, hubo un aumento del 17% del cumplimiento de atenciones de los pacientes. La investigación da como conclusión una interpretación de los procesos que se desarrollan en el servicio de Hospitalización, debido a ello se elaboró la documentación de los procesos, por medio de diagramas de flujo que detallan los pasos que implican cada proceso, así como el personal de salud a cargo de ejecutarlos. Esta información estará al alcance del hospital para normalizar procesos y permitir una comunicación fluida. Asimismo, se implementó un SI que cumple con los requerimientos propuestos y les proporciona de una interacción amigable y sencilla de utilizar, posteriormente el flujo de la atención en el HNR será eficaz.

Nacionales

Chuquilin Llanos y Vásquez Cabrera (2018), realizó la tesis denominada “Implementación de un sistema informático para la gestión de atenciones a los pacientes del puesto de salud Agocucho del distrito de Cajamarca, 2016”, desarrollada en la Universidad Privada Antonio Guillermo Urrelo, en Perú, el principal problema es las deficiencias de la administración de citas del centro de salud Agocucho, por variados problemas que encaran, sobre todo las citas de atención médica, el ingreso de historias clínicas de infantes y mujeres en gestación, la obtención de información, las cuales son de importancia para el área administrativa que trabaja conjuntamente con el centro de salud. La investigación planteo como objetivo principal, Implementar un SI para la administración de citas del puesto de salud Agocucho del Distrito de Cajamarca, utilizaron el diseño pre experimental tipo aplicada con un enfoque cuantitativo, La población es la cantidad de pacientes atendidos se determinó en 900 y con una muestra de 269 del nivel percentil de cumplimiento, Para conseguir los datos de las variables, se realizó las técnicas de entrevista, encuesta y análisis. Entre los primordiales resultados conseguidos se manifiesta que se requiere diseñar y desarrollar un SI para la administración de citas del centro de salud. Los efectos cotejados del pre y post test mencionando que, precedentemente del desarrollo del software el 60% cumplían el servicio de las atenciones a los pacientes del indicador nivel percentil de cumplimiento, y posteriormente del desarrollo del software un 72%, hubo un incremento del 12% de las atenciones a los pacientes del indicador nivel percentil de cumplimiento. El desarrollo del SI se efectuó con éxito para la administración de citas del centro de Salud Agocucho, posibilitando el registro de las citas, búsqueda de historias e informes de los pacientes.

Además, se determinó que la implementación del SI se empleó la metodología XP. Finalmente se tiene la certeza que hay un progreso notorio de los administradores en la gestión de citas del centro de Salud Agocucho, brindando una atención deseable y de calidad, siendo más accesible e infalible el salvaguardado de los datos de atención al paciente, eludiendo el extravío e ingreso no autorizado a la data, produciendo conflictos entre la administración y la atención al paciente.

Este antecedente nos muestra el efecto de como un sistema web puede automatizar distintas actividades de un proceso y este pueda lograr mejorar tiempos de atención utilizando las directrices necesarias para su funcionamiento. Además, menciona a uno de nuestros indicadores y como beneficio un sistema informático al incremento de las atenciones atendidas.

Para Duarte (2016), en el artículo “Sistema web para la administración de atenciones médicas y monitoreo de la unidad de triaje en el Hospital Cayetano Heredia”, expuesta en la Universidad de Lima, Lima. El alcance de la investigación es mecanizar, por medio de un desarrollo web, el proceso del servicio de triaje y la unidad de emergencia de un hospital donde el médico pueda informarse sobre el historial del paciente y acelerar los procedimientos de la unidad de emergencia. El problema fundamental en el hospital es el retraso de la atención hacia el paciente, la obtención de una cita y el nivel de la atención, ya que el actual registro ocasiona cuellos de botella y es aquí en donde existe la pérdida de información; así como, la búsqueda rápida de la información, además el sub área de triaje no posee una comunicación con el área de emergencia lo cual genera que esta desconozca la culminación de la atención del paciente y sobre todo existe un desorden al priorizar los casos que si necesitan una urgencia. Se utilizó un diseño pre experimental tipo aplicada con un enfoque cuantitativo, La población: la cantidad de pacientes atendidos en triaje se determinó en 1100 y con una muestra de 285. El sistema para el desarrollo fue RUP este brinda un buen desarrollo del software y un buen registro de la documentación del sistema. Para la validación se consideró realizar pruebas en tiempo real de la atención en emergencias en donde se comparó los tiempos de la forma tradicional y la utilización del sistema teniendo como resultados obtenidos antes de la implementación del software es de un 77% los cuales cumplían las atenciones de los pacientes atendidos en triaje, y al poner en marcha el sistema web es de un 88%, que hubo un incremento del 11% de las atenciones a los pacientes en triaje. En conclusión, la

automatización del sistema brindo una diferencia de 25 minutos en el tiempo de atención con relación al procedimiento convencional del hospital, se homogenizo la data para incorporarla de acuerdo a las exigencias de los involucrados del servicio de emergencia y sobre todo se encontraron tareas repetitivas y la saturación del servicio de emergencia.

Esta realidad se asemeja en cierta parte al área de emergencia de la clínica san Borja, en donde el problema inicia en el mal funcionamiento de la priorización del triaje y de su registro de la atención del paciente en donde se genera un cuello de botella y surge la demora en tiempo de atención al paciente, sobre todo el rápido accionar que podría tener el médico si es que posee una herramienta que le permita priorizar el triaje de una manera eficaz y eficiente.

Briones Mariño (2018), realizo la tesis denominada “Implementación de un Sistema Web para la Gestión Administrativa en el Centro de Terapia Física y Rehabilitación de la Clínica Good Hope, Miraflores 2018”, en la Universidad Peruana de las Américas, en Perú, el objetivo principal de la investigación es optimizar la Gestión del Proceso de Rehabilitación por medio de un entorno web, debido a que el área de terapia física y rehabilitación no cuenta con un sistema que pueda optimizar sus procesos, La problemática se desarrolla en la realización de una orden médica, se registra manualmente ,no existe seguridad y confiabilidad de la información sobre las atenciones en el centro de terapia física y rehabilitación, se requiere mejorar el proceso de registro de las ordenes médicas desde el despacho médico, de igual forma se evitaría el extravío de la data del paciente. Consecuentemente la realización de itinerario de citas se gestiona manualmente se requiere mejorar transformando el proceso realizando el itinerario de citas para los pacientes; por último, los informes físicos que son empleados para la gestión de asistencias de los pacientes se registran manualmente puesto que al contener la información acopiada en formatos físicos originan el extravío de información y en efecto produce inquietud en los pacientes. Se utilizó un diseño pre experimental tipo aplicada con un enfoque cuantitativo. Se realizó para esta investigación las técnicas de entrevista, encuesta y análisis, La población utilizada de pacientes atendidos se determinó en 1513 y con una muestra de 306 de sus atenciones. Entre los primordiales resultados conseguidos se manifiesta que se requiere diseñar y poner en marcha un Sistema Web para optimizar la Gestión del Proceso de Rehabilitación. Los resultados antes de implementar el sistema web son de un el 66% cumplían el servicio de las atenciones a los

pacientes, y posteriormente de la implementación del sistema web un 79%, hubo un incremento del 13% de las atenciones a los pacientes. Los resultados obtenidos sobre el pase a producción del Sistema Web optimizan la Gestión del Proceso de Rehabilitación de la Clínica Good Hope, se manifiesta hubo un progreso y mejoramiento de forma significativa sobre las problemáticas mencionadas anteriormente y progreso significativo en los procesos que se desarrollaban manualmente. En relación a la emanación de órdenes médicas hubo una mejoría relevante, además el ingreso de los datos que el doctor realiza de las atenciones, todo esto y otros requerimientos se solucionaron con el sistema web desarrollado.

Esta investigación brinda un gran aporte ya que se asemeja a la realidad problemática que tiene la clínica Sanna San Borja tomando como modelo para optimizar la atención hacia el paciente y para el proceso de emergencia médica.

Huamán Zamudio (2017), realizó la tesis denominada “Implementación de un sistema web de historial Clínico para la mejora de los procesos de historias clínicas del centro de Salud de la Provincia de Cañete”, desarrollada en la Universidad Peruana de las Américas, en Perú, el objetivo de la investigación es precisar la influencia del sistema web de historial Clínico con el mejoramiento de procesos de historias clínicas del centro de salud. El centro de salud posee diversos problemas en las áreas de admisión y archivos, atribuyendo una observación previa, se evidencia que existe un retraso excesivo de atención en relación a la hora de ingreso y la finalización, este problema compromete a las distintas áreas relacionadas al proceso de atención al paciente con respecto a la historia clínica. Se empleó un diseño pre experimental tipo aplicada con enfoque cuantitativo para la investigación. La población de pacientes atendidos se determinó en 600 y con una muestra de 238 del nivel porcentual de pacientes atendidos, Para conseguir los datos de las variables, se realizó las técnicas de entrevista, encuesta y análisis. Los resultados indican que se requiere desarrollar un sistema web para optimizar los procesos de registro de historia clínica. Los resultados antes del desarrollo del sistema web es de 75% que cumplían las atenciones a los pacientes del indicador nivel porcentual de pacientes atendidos, y posteriormente del sistema web es de 90%, hubo un incremento del 15% de las atenciones a los pacientes del indicador nivel porcentual de pacientes atendidos. La presente investigación llega a la conclusión que en relación al desarrollo del sistema web reduce los incidentes en la administración de historias clínicas del centro de salud, puesto

que se encontraron diversos problemas como el extravió de historias clínicas, demora en la atención, la aglomeración en relación a las colas diarias de los pacientes para ser atendidos, Actualmente con el sistema web desarrollado se han notado mejoras, como el extravió de informes de historia clínica, mejora en los tiempos de atención entre las áreas de admisión y caja.

De este trabajo previo se tomó en cuenta la variable independiente y ayudó reforzar los indicadores con respecto a la mejora del flujo de atención al paciente en el proceso de emergencia.

1.3. Teorías relacionadas al tema

Variable dependiente: Proceso de emergencia médica

Para Cárdenas y Roca (2017), menciona al proceso de emergencia médica como “las habilidades, actitudes y los conocimientos necesarios para la previsión, el diagnóstico y terapia de aspectos urgentes donde el tiempo es vida”. (p. 101)

Según Serrano et al (2015), define que el proceso de emergencia médica como “aquella coyuntura inaplazable que, al ser evaluada por la atención inicial, establece si está en peligro la salud del paciente, por lo que requerirá un control próximo, que será examinado en el área de auxilio o emergencia”. (p. 12).

Según Jiménez (2014), define que “el proceso de emergencia médica es aquella circunstancia de peligro inminente a la vida que impulsa a desarrollar procedimientos y recursos particulares y requieren un método rápido para salvaguardar la supervivencia del paciente y en algunas ocasiones brindar un diagnóstico con prontitud”. (p. 10).

Fases del proceso de emergencia médica

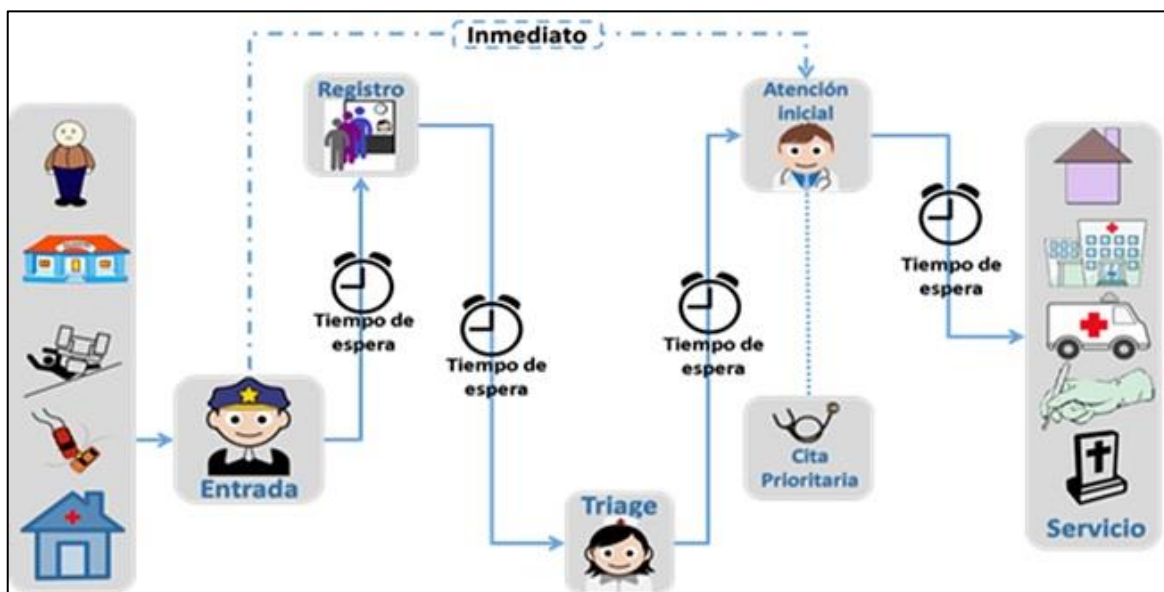
Según Restrepo et al. (2018, p.4), menciona en su investigación que las unidades de emergencias médicas se representan con un esquema (figura 5) con los distintos eventos de la atención los cuales divide de la siguiente manera:

1. **Entrada:** Es la actividad en el que el paciente ingresa al área de admisión emergencia y detalla su ingreso.
2. **Registro:** Es la actividad del registro o admisión del paciente en donde se brinda documentación e información básica.

3. **Triage:** Es la actividad para clasificar al paciente según su estado de salud el cual posee dos modelos
 - Modelo Australiano: Este protocolo emplea 4 niveles de prioridad.
 - Modelo Manchester: Este protocolo emplea 5 niveles de prioridad.
4. **Atención inicial:** Es la actividad de la atención médica acompañada de la atención asistencial de la enfermera.
5. **Cita prioritaria:** Actividad a la que refiere a exámenes de apoyo al diagnóstico.
6. **Canal de salida:** Es la actividad de dar el alta a la atención del paciente o la derivación de este a otras zonas.

Además, Restrepo et al. (2018), define que “en ciertas etapas del proceso se muestran periodos de demora en el registro de admisión, triaje, atención inicial o alta médica, donde podrían generarse cuellos de botella, [...]”. (p. 4).

Figura 5 Proceso de Emergencia Medica



Fuente: Restrepo et al. (2018)

Entrada

Según Del Rio (2014), define que “es la actividad de acoger al usuario, familia y acompañantes, respondiendo a sus requerimientos en el marco de una relación basada en el respeto mutuo”. (p. 3).

Registro

Merlo et al. (2014), define al “el área de admisión como el acceso de entrada de los usuarios en las distintas áreas en un hospital, esta área tiene como tarea admitir y dar la sensación de confort al paciente y familiares [...]”. (p. 5)

Triaje

Asimismo, Garcia et al (2018), define al triaje “como el proceso de evaluación clínica básica que clasifica y atiende a los pacientes según el grado de urgencia que presenten dando la prioridad asistencial a los que más necesitan”. (p. 69).

Según Bibiano (2018, p. 24), define que Faculta un control del peligro clínico y una evaluación de los pacientes por medio de su nivel de emergencia para realizar una administración conveniente y tener una solidez en la atención médica en donde las actividades del triaje son las siguientes:

- ✓ Reconocer a los pacientes con prioridad urgente.
- ✓ Clasificar el nivel de prioridad.
- ✓ Realizar la revaloración de los pacientes que están en espera.
- ✓ Retribuir el box correspondiente a la gravedad de la dolencia del paciente.
- ✓ Brindar la información necesaria para la atención médica.
- ✓ Evitar las congestiones y cuellos de botella durante la atención.

Para Mesquita, Pavlicich y Luaces (2016), define que “el triaje en el departamento de emergencias clasifica a los pacientes en niveles según la prioridad en la atención”. (p. 107).

Atención inicial

Según los autores Hamui, Paulo y Hernández (2018), define que “es la actividad principal y el marco primordial de la profesión médica; no obstante, es el proceso de la enfermedad-curación, lleva a reflexionar sobre varios contextos socioculturales y el impacto en la comunicación del médico y paciente”. (p. 32).

Cita prioritaria

Hernández et al. (2017), define que “son los elementos de apoyo al médico en su práctica clínica, brindando información para la diagnosis, la previsión o la terapia de enfermedades”. (p. 13).

Canal de salida

Según la SUCESO (2018), define que “se infiere que la alta médica es el documento que brinda el médico sobre la finalización del tratamiento médico, recuperación y otros casos que suceden durante el tratamiento, para conseguir la sanación del enfermo”. (p. 12)

Dimensión e indicadores del proceso de emergencia medica

Dimensión: Triage

Indicador 1: Nivel porcentual de pacientes atendidos en triaje

Según Alonso Cristóbal, Alonso Gonzáles y Cabral (2016), “Se precisa como la proporción de pacientes con este periodo menor a 10 minutos entre el total de pacientes a clasificarse. El Sistema Estructurado de Triage establece un patrón que el resultado debe ser mayor al 85%”. (p. 18).

Figura 6 Nivel porcentual de pacientes atendidos en triaje

$$NPCA = \frac{CPC}{CTP} * 100\%$$

Fuente: Alonso Cristóbal, Alonso Gonzáles y Cabral (2016)

Dónde:

NPCA= Nivel porcentual de pacientes atendidos en triaje

CPC= Cantidad de pacientes atendidos en triaje

CTP= Cantidad de pacientes admitidos

Indicador 2: Nivel percentil de cumplimiento

Según Alonso Cristóbal, Alonso Gonzáles y Cabral (2016), “Se precisa que un 90% de los atendidos en triaje deben ser evaluados en menos o igual a 2 horas desde su priorización en el área de triaje y el 100% en menos o menor a 4 horas”. (p. 18).

**Figura 7 Formula Nivel percentil de cumplimiento
indicador 2**

$$NPC = \frac{PAP}{TPAT} * 100\%$$

Fuente: Alonso Cristóbal, Alonso Gonzáles y Cabral (2016)

Dónde:

NPC= Nivel percentil de cumplimiento

PAP= Pacientes atendidos por prioridad

TPAT= Cantidad de pacientes admitidos

Variable independiente: Sistema web utilizando el protocolo de Manchester

Sistema Web

Para Ramos, A. y Ramos, M. (2014), definen sobre “El sistema web como una herramienta que brinda información, desarrolla transacciones, que se puede transigir hacia otro suministro por medio de la web y se puede notificar a través de reglamentos de internet, y están desarrollados para autorizar mensajes de un servicio a otro”. (p, 10).

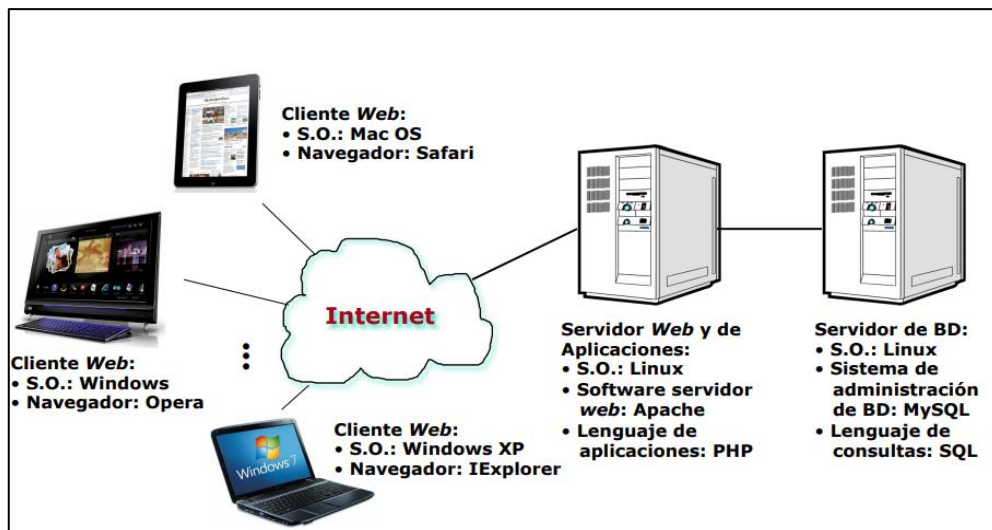
Para Báez (2014), define a un “sistema web como aquellos recursos que los clientes y/o internos al negocio pueden acceder a un servicio web mediante internet o intranet utilizando un web browser.” (p. 10).

Según Beltrán y Sevillano (2013), define a los sistemas web “como aplicaciones de fácil acceso, a disimilitud de una aplicación tradicionales, donde cambian data constantemente. Los sistemas web utilizan links para que puedan acceder a ellas, las cuales tienen diferente modo de acceso y algunos generan diferentes medios para acceder” (p.22).

Arquitectura de un sistema web

Para Cubillos (2018, p. 3), la estructura del sistema web lo define como: “Servidor: realiza la ejecución de un conjunto brindando tareas a los usuarios. Usuario: llama tareas del servidor. [...] Presentación: programa que enseña apropiadamente los efectos de un software. Software: programas que brindan soluciones para el cliente. Gestión de la data: control de la data que apoyan a los programas”:

Figura 8 Arquitectura Web



Fuente: Cubillos (2018)

Metodología para el desarrollo del software – Sistema Web

OOHDM

Para Molina et al. (2018), define que “este método posibilita produce software en entorno web desde el uso de paradigmas específicos tales como: conceptual, navegación e interfaz de usuario teniendo como alcance reducir y realizar diseños más eficientes para los programas” (p. 8).

Metodología UWE

Según Atahuichi (2014), define que “es un método de producción de software en entorno web, empleado en la especialidad web, dedicado a la sistematización y personalización, es decir realizar sistemas adaptativos. [...] sus etapas concuerdan con sugerencia de UML, pero adaptada a la web”. (p. 26)

Metodología XP

Según Ramos et al. (2017), define que “xp se estima como un procedimiento ligero de producción de software. [...] viene transformando para solucionar los inconvenientes de al entregar el software y poder lograr las exigencias de un negocio que siempre transmuta”. (p. 211).

Selección de la metodología para el sistema web

Se utilizó algunos criterios evidenciados en la tabla N° 1 para seleccionar la metodología de desarrollo para implementar el sistema web, estos poseen características necesarias en donde tendrán un puntaje significativo el cual será evaluado en el juicio de expertos con la ayuda de un experto en metodologías de desarrollo (**VER ANEXO 6**).

Tabla 1 Descripción de criterios para la selección de metodología

Criterios	Descripción
Resultados rápidos	Se requiere que la metodología brinde los resultados requeridos y rápidos de acuerdo a los requerimientos funcionales que se solicitaran para su desarrollo.
Adaptabilidad	El apoyo de la metodología seleccionada y diseñada de diferentes patrones para la implementación del software permitirán que el sistema sea desarrollado para hacerlo portable.
Asegura un software de calidad	La metodología seleccionada generara un software de calidad debido al levantamiento de requerimientos, análisis y diseño de cada diagrama para su desarrollo.

Desarrollo orientado a objetos	La metodología seleccionada trabaja de la mano con la programación orientada a objetos para su óptimo desarrollo.
Implementa las necesidades del sistema	La metodología seleccionada brinda las directrices necesarias para el análisis y diseño respectivo para la implementación del software.
Integra eficientemente todas las fases del ciclo del software	La metodología seleccionada brinda las fases necesarias para un óptimo desarrollo del sistema web, desde los requisitos hasta su implementación y mantenimiento.
Desarrollo iterativo e incremental	La metodología seleccionada permite generar los cambios necesarios ya que tiene la característica de la adaptabilidad.
Desarrollo a base de la comunicación con el usuario y retroalimentación	La metodología seleccionada permite tener en contacto al usuario final y conocer sus diversas actividades que realizara en el sistema web.

Fuente: Elaboración Propia

Evaluación de la metodología para el sistema

La presente investigación fue validada por 3 expertos en asesoramiento de metodologías de la cual se tomó del formato de criterio de expertos (**ANEXO 6**) se puede observar en la tabla N° 2 la comparación entre las distintas metodologías aplicadas para la investigación y la cual será elegida por el experto en metodología de desarrollo.

Tabla 2 Matriz de evaluación de las metodologías por expertos

Expertos	Grados	OOHDM	UWE	XPX
Gálvez Tapia, Orleans	Magister	24	18	16
Cueva Villavicencio Juanita Isabel	Magister	21	20	19
Díaz Reátegui, Mónica	Doctora	24	15	14
Promedio		69	53	49

Fuente: Elaboración Propia

El puntaje expuesto por el juicio de expertos en la tabla N°2 opta como metodología de desarrollo para la investigación a la metodología OOHDM obteniendo un puntaje de 69 puntos.

Metodología Seleccionada: OOHDM

Según Santiago Mariño (2017) manifiesta que “[...] La metodología OODHM sugiere el desarrollo de aplicaciones hipermedia por medio de un proceso compuesto por cinco etapas: obtención de requerimientos, diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación.” (p. 1).

Características de OODHM

Según Santiago Mariño (2017) manifiesta que “La primera de ellas es que OODHM está basada en el paradigma de la orientación a objetos. [...] fundamentalmente OODHM toma como partida el modelo de clases que se obtiene en el análisis del proceso unificado de UML.” (p. 2).

Fases de la metodología OODHM

Obtención de requerimientos

Según Santiago Mariño (2017) manifiesta que “La adquisición de requerimientos es la etapa fundamental, dado que es adonde se recopila los datos, entretanto mayor sea el grado de profundidad de la recolección, mejor posibilidad de obtener una aplicación de acuerdo a las expectativas de los usuarios.” (p. 5).

Diseño conceptual

Según Santiago Mariño (2017) manifiesta que “En este punto se construye un esquema conceptual representado por los objetos del dominio, [...]. En OODHM, el esquema conceptual está construido por clases, relaciones y subsistemas.” (p. 7).

Diseño Navegacional

Según Santiago Mariño (2017) manifiesta que “En OOHDM, un modelo navegacional es construido como una vista sobre un diseño conceptual. Admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios [...]” (p. 9).

Diseño de Interfaz Abstracta



Según Santiago Mariño (2017) manifiesta que “Esto significa definir la forma en la cual los objetos navegacionales pueden aparecer, como los objetos de interfaz activaran la navegación y el resto de funcionalidad de la aplicación” (p. 12).

Implementación

Según Santiago Mariño (2017) manifiesta que “Es la fase en donde se implementa el diseño. Hasta entonces, los modelos son diseñados independientemente de la implementación; en este punto se especifica el ambiente en donde se pondrá en marcha la aplicación.” (p. 14).

Comparativo de Protocolos de Triage

Tabla 3 Comparativa de protocolo de Triage

	M TS	SET- MAT
Prioridades	Prioridad: (I, II, III, IV, V) Inmediata, Muy Urgente, Urgente, Normal y No Urgente	Prioridad: (I, II, III, IV, V) Reanimación, Emergencia, Urgente, Menos Urgente y No Urgente
Colores	Prioridad: (I, II, III, IV, V) Rojo, Naranja, Amarillo, Verde y Azul	Prioridad: (I, II, III, IV, V) Azul, Rojo, Naranja, Verde y Negro
Tiempos de espera	Prioridad: (I, II, III, IV, V) 0 minutos, 10 minutos, 60 minutos, 120 minutos y 240 minutos	Prioridad: (I, II, III, IV, V) inmediato, 7 minutos, 30 minutos, 45 minutos y 60 minutos
Origen		
Aparición	1994	2000
Basado	Sintomatología y signos	Categorías sintomatológicas
Motivos de atención	52 motivos agrupado en 5 categorías	617 motivos agrupados en 32 categorías sintomáticas y 14 subcategorías
Discriminador es	Signos vitales, tiempo de síntoma, riesgo vital y nivel de dolor.	Signos vitales
Personal de traje	Enfermera	Enfermería no excluyente

Fuente: PASIES, Laura (2019)

Protocolo de Manchester

Según Díaz, Álvarez y Pardillos (2014) define que protocolo de Manchester “Es un sistema de triaje denominado Manchester Triage Systems que utilizan muchos hospitales

cuando se produce la entrada de un paciente a la unidad de emergencias. [...] sirve para organizar a los pacientes antes de ser diagnosticados y tratados” (p.435).

Por otro lado, Soler y Raurich (2015) definen que “El protocolo de Manchester es una técnica con criterios que clasifica al paciente que llega a emergencias en uno de 52 motivos de consulta. La clasificación del paciente es según su estado en un código de colores [...]” (p.134).

Asimismo, Caballero y Zapata (2019) define que Los sistemas SET y MTS actualmente son los más utilizados en España y conforman de cinco niveles de prioridad, que se constituyen de la siguiente manera:

1. Discriminadores clínicos, clases significativas, amplia o rígida. Son grupos de signos o síntomas que el médico debe identificar y descifrar por medio del diagnóstico que le posibilita categorizar en el nivel exacto y prioridad de atención. El protocolo de Manchester cataloga al paciente en los 52 motivos posibles.
2. Discriminadores y grado del nivel de emergencia: peligro a la salud, manifestos fisiológicos, periodo de alteración, grado de dolor, motivos de lesión, etc. (p.955).

Figura 9 Niveles de Clasificación Manchester

Color	Clasificación	Tiempo máximo para atender
Rojo	Emergencia - Existe riesgo inmediato de la vida del paciente y debe ser atendido inmediatamente	0 minutos
Naranja	Muy urgente - Existe riesgo para la vida del paciente y debe ser atendido cuanto antes	Hasta 10 minutos
Amarillo	Urgente - No se considera una emergencia, pero el paciente necesita pasar por una evaluación	Hasta 60 minutos
Verde	Poco urgente - Es considerado un caso menos grave, el paciente puede aguardar atención o ser encaminado a otro servicio de salud	Hasta 120 minutos
Azul	No es urgente - es el caso más simple, el paciente puede aguardar por atención o ser encaminado a otro servicio de salud	Hasta 240 minutos

Fuente: Caballero y Zapata (2019)

1.4. Formulación del problema

Problema General

PG: ¿De qué manera influye un sistema web utilizando el protocolo de Manchester en el proceso de emergencia médica en la Clínica Sanna San Borja?

Problemas Específicos

PE 1: ¿Cómo influye un sistema web utilizando el protocolo de Manchester en el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en la Clínica Sanna San Borja?

PE 2: ¿Cómo influye un sistema web utilizando el protocolo de Manchester en el nivel percentil de cumplimiento en el proceso de emergencia médica en la Clínica Sanna San Borja?

1.5 Justificación de estudio

Justificación Institucional

Según Aguilera et al (2017) nos comenta, que “La mejora continua en la actualidad no es más un elemento o un canal para lograr un alcance, sino que la evolución y el progreso tecnológico han logrado obtener la optimización como parte central de la planificación y organización de las empresas”. (p. 10)

La investigación contribuirá al lograr los alcances que se propone la empresa ya que desea alcanzar una atención de calidad hacia sus pacientes y además lograr tener una mejor imagen para la clínica. Sobre todo, el sistema web brindara el funcionamiento correcto utilizando el protocolo de Manchester que da un punto de calidad hacia la clínica y su funcionamiento.

Justificación Económica

Según Diez, Martin y Montoro (2014) plantea que “El programa es factible, a partir del plano económico, si es capacitado de lograr servicios y productividad” (p. 348).

Actualmente en la clínica para atender en el proceso de emergencia consta de 10 personas entre personal médico profesional teniendo aproximadamente un sueldo de S/.5 000.00 soles c/u, 10 enfermeros y/o enfermeras teniendo aproximadamente un sueldo de S/.3

500.00 c/u, 5 internistas teniendo aproximadamente un sueldo de S/.2 000.00 soles c/u, 4 admisionistas teniendo aproximadamente un sueldo de S/.1 500.00 soles c/u, obteniendo en total un desembolso mensual de S/. 101 000.00 soles y anual se desembolsa en el proceso de mantenimiento un aproximado de S/. 1 212 000.00 soles. Sin embargo, al poner en marcha el sistema web, no se tendrán que solicitar personal adicional para el mantenimiento, ya que solo quedaran atendiendo 2 admisionistas, 2 internistas, suscitando un desembolso mensual de S/. 9 000.00 soles y anual sería S/. 108 000.00 soles. En síntesis, al poner en marcha el sistema en entorno web que permita monitorizar el proceso del flujo del proceso de emergencia por lo que se demuestra económicamente, obteniendo beneficios y el reintegro de la inversión.

Justificación Operativa

Para Vélez (2017) nos plantea, que “la construcción de las ciencias tecnológicas de información ha contribuido [...], por este motivo las empresas invierten para mejorar un proceso de negocio específico ya que supone una importante reducción de costes empresariales que se traducen a un gran ahorro”. (p. 18)

Implementar el sistema web permitirá disminuir tiempos y cuellos de botella en el proceso, a su vez le otorgará cumplir con los objetivos establecidos permitiendo a la parte medica tener acceso a la web las 24 horas y sobre todo accesos desde cualquier ubicación para desarrollar sus actividades médicas.

Justificación Tecnológica

Según Sánchez (2017) nos plantea, “[...]. La tecnología constituye el principal determinante de comportamiento y los cambios organizacionales donde la innovación es el motor principal para un cambio de la organización.” (p. 157).

Implementar un sistema web utilizando el protocolo de Manchester no solo brindará la automatización de la prioridad hacia el paciente, sino que también optimizará la calidad del servicio y le permitirá a la Clínica Sanna San Borja ser más reconocida en el rubro de salud.

1.6 Hipótesis

Hipótesis General

HG: El sistema web utilizando el protocolo de Manchester mejora el proceso de emergencia médica en la Clínica Sanna San Borja.

Hipótesis Específicas

HE 1: El sistema web utilizando el protocolo de Manchester aumenta el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en la Clínica Sanna San Borja.

HE 2: El sistema web utilizando el protocolo de Manchester aumenta nivel percentil de cumplimiento en el proceso de emergencia médica en la Clínica Sanna San Borja.

1.7. Objetivos

Objetivo General

OG: Determinar la influencia del sistema web utilizando el protocolo de Manchester en el proceso de emergencia médica en la Clínica Sanna San Borja.

Objetivos Específicos

OE 1: Determinar la influencia del sistema web utilizando el protocolo de Manchester en el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en la Clínica Sanna San Borja.

OE 2: Determinar la influencia del sistema web utilizando el protocolo de Manchester en el nivel percentil de cumplimiento en el proceso de emergencia médica en la Clínica Sanna San Borja.

II. MÉTODO

2.1. Diseño de la investigación

Para Bonetto (2016) el método hipotético deductivo “inicia desde el planteamiento de la hipótesis, infiriendo esto como deducción o aseveración temporal como inicio de una porción para el estudio [...]” (p. 2).

La investigación empleará el método hipotético deductivo, ya que al recolectar la información y la distinción de resultados se podrá confirmar las hipótesis expuestas.

Tipo de investigación

Según Muñoz (2015) define que la investigación explicativa “analiza las relaciones casuales o las condiciones en que un fenómeno se produce, es la más profunda, pues explica la razón, el porqué de las cosas.” (p. 109).

La investigación es un estudio explicativo, porque se busca exponer los factores o motivos del problema que originan los diferentes fenómenos en la presente investigación, ya que busca poner a prueba sus hipótesis en relación a sus variables estudiadas.

Según Lerma (2016) define que la investigación experimental “tiene como alcance demostrar la conexión causa-efecto sobre dos elementos o más. El observador altera deliberadamente la condición de algunos de los objetos de instrucción, insertando y maniobrando el procedimiento o mediación que aspira valorar.” (p. 224).

El estudio es de tipo experimental, ya que está orientada a analizar la variable dependiente del proceso de emergencia médica realizando el análisis por medio de un Pre – test y un Post – test de los indicadores y así poder analizar los resultados.

Según Muñoz (2015) definen que la investigación aplicada, “es recomendable cuando el tema, la idea o el proyecto de investigación busca la solución de problemas prácticos y concretos [...]” (p. 112).

La investigación es del tipo aplicada – experimental, a causa de poner en funcionamiento un sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica, el cual permitirá resolver la problemática presentada en la Clínica Sanna San Borja.

Diseño de investigación

Según Cruz, Olivares y Gonzáles (2014) definen que el diseño pre experimental “es un diseño de pre y post prueba con un conjunto específico. A una agrupación se le atribuye una comprobación anticipada a la incitación o trato experimental luego se gestiona el trato y por último se le emplea una corroboración posterior al trato.”. (p. 130).

El diseño de investigación es pre experimental porque se aspira desarrollar un sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica en la clínica empleando el modo pre-test y post-test.

Figura 10 Diseño de investigación

$$G : O_1 : X : O_2$$

Fuente: Elaboración propia

Dónde:

G: Grupo experimental: Este grupo muestral es al cual se realizó una medición para calcular la dimensión de triaje y medir el nivel percentil de cumplimiento y el nivel porcentual de pacientes atendidos en triaje en cada fase del proyecto.

O1: Pre – Test: Se midió a un grupo experimental previo a la implementación del sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica. Esta medición servirá para comparar con un Post - Test

X: Experimento (Implementación del Sistema web): Aplicando el sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica. Por medio de las evaluaciones (Pre – Test y Post – Test), los cuales pueden medir si el sistema web utilizando el protocolo de Manchester mejoro y si genero variaciones en el proceso de emergencia médica .

O2: Post – Test: Se midió a un grupo experimental previo a la implementación del sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica. Se realizaron mediciones para comparar y ayuden a definir el nivel porcentual de pacientes atendidos en triaje y el nivel percentil de cumplimiento; previo y posteriormente a la implementación del sistema web utilizando el protocolo de Manchester.

2.2 Variables, Operacionalización

Definición Conceptual

Variable independiente (VI): Sistema web utilizando el protocolo de Manchester

Según Cardador (2014) definen un “un sistema web es un programa de software el cual se desarrolla por un cierto lenguaje de programación, sostenido por navegadores web y en donde el usuario interacciona con un computador central”. (p. 75).

Según CEP (2017) definen al “protocolo de Manchester como la escala que categoriza al paciente que ingresa a emergencias en 52 razones distintas según signos y síntomas, con discriminantes, después realiza la clasificación del paciente en cinco clases, [...]” (p. 557)

Variable dependiente (VD): Proceso de emergencia medica

Según CEP (2017) definen que el proceso de emergencia médica es “las actividades de la medicina de emergencias se desarrolla sobre una emergencia o urgencia médica, determinadas en lesiones o enfermedades que proyectan un amedrentamiento continuo sobre la salud del paciente y ayuda no debe ser retardada”. (p. 33)

Definición operacional

Variable independiente (VI): Sistema web utilizando el protocolo de Manchester

El sistema web utilizando el protocolo de Manchester permitirá el registro de los datos paciente y la asignación de una orden de atención, también permitirá la priorización y registro de la atención del paciente desde el área de triaje para que esta sea rápida y precisa, además permitirá el registro de la atención médica para que esta sea oportuna y rápida de procesar.

Variable dependiente (VD): Proceso de emergencia médica

Consiste en las actividades médicas para la asistencia de la salud del paciente que se encuentra en peligro de vida el cual debe ser resuelto al instante y no debe ser afectada ni manipulada por otro personal que no sea el médico especialista.

Tabla 4 Operacionalización de las variables

Variable	Definición Conceptual	Definición Operacional	Dimensión	Indicadores	Escala de Medición
Variable Independiente X_1 : Sistema web utilizando el protocolo de Manchester	<p>Un sistema web es una aplicación que se utiliza en un entorno web y se codifica en un lenguaje de alto nivel, el medio para ingresar son los navegadores web el cual permite el usuario interactúe con un servidor web.</p> <p>El protocolo de Manchester es la escala que cataloga al paciente que ingresa al servicio de emergencia en 52 motivos y sus discriminantes, después realiza la clasificación del paciente por prioridad, estas se identifican por colores y tiempo de atención. Esta distribución permite la evaluación inmediata del paciente por su gravedad</p>	El sistema web utilizando el protocolo de Manchester permitirá el registro de los datos paciente y la asignación de una orden de atención, también permitirá la priorización y registro de la atención del paciente desde el área de triaje para que esta sea rápida y precisa, además permitirá el registro de la atención médica para que esta sea oportuna y rápida de procesar.			
Variable dependiente Y_1	Actividades de la medicina de emergencias, que atienden lesiones o enfermedades que amenacen la vida del paciente y de atención inmediata.	Consiste en las actividades médicas asistenciales en donde el paciente se encuentra en peligro de vida y el cual debe ser resuelto al instante y no debe ser afectada ni manipulada por otro personal que no sea el médico especialista.	Triage	Nivel porcentual de pacientes atendidos en triaje	Porcentaje
				Nivel percentil de cumplimiento	Porcentaje

Fuente: Elaboración Propia

Tabla 5 Indicadores

Variable	Indicador	Descripción	Instrumento	Unidad de Medida	Formula
Variable Dependiente Proceso de emergencia médica	I1= Nivel porcentual de pacientes atendidos en triaje.	Es el nivel porcentual de los pacientes admisionados que están atendidos en triaje.	Ficha de registro	Unidad	$NPCA = \frac{CPC}{CTP} * 100\%$ <p>Dónde: NPCA= Nivel porcentual de pacientes atendidos en triaje. CPC= Cantidad pacientes atendidos en triaje. CTPA= Cantidad de pacientes admitidos</p>
	I2= Nivel percentil de cumplimiento	Es el nivel porcentual de los pacientes atendidos en triaje y priorizados en cada nivel del protocolo de Manchester y que son atendidos en emergencia en el tiempo indicado.	Ficha de registro	Unidad	$NPC = \frac{PAP}{TPAT} * 100\%$ <p>Dónde: NPC= Nivel percentil de cumplimiento. PAP= Pacientes atendidos por prioridad. TPAT= Total pacientes admitidos</p>

Fuente: Elaboración Propia

2.3 Población y Muestra

Población

Para Hernández, Fernández y Batista (2014), la población es el “grupo de todas las coyunturas que comparten las mismas características, las poblaciones están posicionadas inteligiblemente a base de sus peculiaridades de capacidad de espacio y periodo” (p. 321).

La población para la siguiente investigación se determinó 1493 Registros de triaje, agrupados en 20 reportes del Nivel porcentual de pacientes atendidos en triaje y 1211 Registros de triaje, agrupados en 20 reportes del Nivel Percentil de cumplimiento.

Muestra

Según Hernández, Fernández y Batista (2014), la muestra es una “porción de una población de la cual se recoge los datos y que tiene que ser característico del grupo” (p.322).

Para tener la obtención de la muestra se realizará la siguiente fórmula:

$$n = \frac{Z^2 * N}{Z^2 + 4N * (EE)^2}$$

N= Tamaño de muestra

N= Población

Z= Nivel de confianza al 95% (1.96)

EE= Error estimado al 5%

Para el Indicador 1 Nivel Porcentual de pacientes atendidos en triaje:

$$n = \frac{(1.96)^2 * (1493)}{(1.96)^2 + 4 * (1493) * (0.05)^2}$$

$$n = 306$$

Empleando la fórmula se establece que, para la investigación, serán consignados 306 Registros de triaje agrupados en 20 reportes del nivel porcentual de pacientes atendidos en triaje durante 1 mes.

Para el Indicador 2 Nivel Percentil de cumplimiento:

$$n = \frac{(1.96)^2 * (1211)}{(1.96)^2 + 4 * (1211) * (0.05)^2}$$

$$n = 292$$

Empleando la fórmula se establece que, para la investigación, serán consignados 292 Registros de triaje agrupados en 20 reportes del nivel percentil de cumplimiento durante 1 mes.

Muestreo

Según Martínez (2015), define que el muestreo es “la distinción de aquellos sujetos que permitan la obtención de conocimientos útiles para la resolución de los objetivos de la investigación y que simultáneamente, garantice la representatividad de la misma respecto al comportamiento general”. (p. 166).

Para esta investigación se ha seleccionado como muestra para el indicador de nivel porcentual de pacientes atendidos en triaje de 306 registros de triaje y para el nivel percentil de cumplimiento una muestra de 292 registros de triaje.

Muestreo probabilístico

Según Dueñas (2015), define al muestreo probabilístico como “la técnica de muestreo que permite que las muestras sean recogidas en un proceso en el que los individuos de la población poseen la misma oportunidad de integrar la muestra”. (p. 128).

En la presente investigación en ambas poblaciones tienen las mismas oportunidades para pertenecer a la muestra ya que poseen las mismas características que permitirán el objetivo de la investigación.

Tipo de Muestreo Probabilístico: Aleatorio Simple

Para Acosta (2015), menciona que el muestreo aleatorio simple “está centrado en la elección de la muestra al azar de una población y son seleccionados al azar la totalidad de los elementos que contiene la muestra” (p.42).

El muestreo de este estudio es del tipo aleatorio simple puesto a que cada parte tiene posibilidad de ser elegido. Para lo cual se elegirá el muestreo aleatorio simple.

2.4. Técnicas e instrumento de recolección de datos, validez y confiabilidad

Técnicas de recolección de datos

Para Villafuerte (2014), menciona que “Son instrucciones metodológicas y sistemáticas que se ocupan de llevar en práctica y desarrollar procedimientos de investigación y la información es sencilla de agrupar en forma inminente, [...]” (p.23).

Este estudio ha tomado la técnica para el acopio de datos a la técnica de fichaje.

Técnica: fichaje

Para Namakforoosh (2013), define que “El fichaje es una técnica que apoya a diversas técnicas utilizadas en investigación científica; se basa en insertar la data que se va extrayendo en las herramientas nombradas fichas, estas correctamente desarrollados y organizadas [...]” (p.100).

En la presente investigación se van a utilizar fichas de registro, en donde se recogerá la data de las repercusiones manifestadas diariamente en la clínica Sanna-San Borja, logrando así una oportuna recolección de datos, con el fin de medir las variables presentadas en el estudio.

Instrumento de recolección de datos

Según Vigil (2018), define que “Es un medio que emplea el investigador para examinar la data o base de datos sobre los elementos de la investigación. Y es adecuado cuando registre data aparente que interprete realmente las ideas, las cuales deben agruparse en tres condiciones indispensables: confiabilidad, autenticidad y objetivismo.”

En la investigación se ha tomado como instrumento a las fichas de registro para reunir los datos necesarios para el proyecto.

Ficha de registro

Según Namakforoosh (2013) agrega que “Permite organizar y catalogar la data consultada y su finalidad es denotar las veces que acontece un fenómeno o acontecimiento. Desde la perspectiva de la medición, son diversos artefactos existentes y los tipos de medios para formalizar un registro” (p.101).

En esta investigación se tomó como instrumento de recolección de datos fichas de registro con la finalidad de evaluar el proceso de emergencia médica de la Clínica Sanna - San Borja, con la manera de que ayude a realizar dicha medición a cada indicador empleando un Pre-test y una Post-Test.

Validez

Según Hernández, Fernández y Baptista (2014), define que “La validez, en términos generales, se refiere al grado en que un instrumento mide realmente la variable que pretende medir” (p.201).

Para la presente investigación dicha validez aplicada para el instrumento se realizó mediante un juicio de expertos.

Validez de contenido

Según Hernández, Fernández y Baptista (2014), define que “La validez de contenido se refiere al grado en que un instrumento refleja un dominio específico de contenido de lo que se mide” (p.201).

Para la presente investigación se utilizaron fichas de registros para medir el contenido según los registros de triaje agrupadas por cada indicador.

Validez de Criterio

Según Hernández, Fernández y Baptista (2014), define que “La validez de criterio de un instrumento de medición se establece al comparar sus resultados con los de algún criterio externo que pretende medir lo mismo” (p.202).

En esta investigación se utilizaron fichas de registro de meses anteriores para hacer las comparaciones de resultados respectivas de un mes a otro.

Validez de Constructo

Según Ruiz (2013), agrega que “Esta validez interesa cuando se quiere usar el desempeño de los sujetos con el instrumento para inferir la posesión de ciertos rasgos. Para estudiar este tipo de validez es necesario que exista una conceptualización clara del rasgo estudiado basado en una teoría determinada” (p.235).

En esta investigación se utilizó una formula por cada indicador, basado en los registros de triaje.

Tabla 6 Validación de Instrumento

Experto	Grado	Ficha de registro: Nivel porcentual de pacientes atendidos en triaje	Ficha de registro: Nivel percentil de cumplimiento
Mg. Cueva Villavicencio Juanita Isabel	Magister	85%	85%
Mg. Gordillo Humanchumo Luis	Magister	78%	80%
Mg. Ormeño Rojas Robert Eduardo	Magister	80%	80%
PROMEDIO		81%	81.67%

Fuente: Elaboración propia

Se evidencia que se realizó la validez de recolección de datos a través del juicio de expertos en la tabla N°6, se. (**VER ANEXO 6**), obteniendo como resultado para el nivel porcentual de pacientes atendidos en triaje a un 81%, esto manifiesta que el instrumento de medición a emplear es apto y el nivel percentil de cumplimiento es de 81.67%, esto manifiesta que el instrumento de medición a emplear es apto.

Confiabilidad del instrumento

Según Cervantes (2013) indica que “[...]. Se puede decir que la confiabilidad es una medición del error que puede generar un instrumento al ser inestable y aplicarse en diferentes ocasiones.” (p. 28)

En esta investigación y en la validación de los expertos se verifica que la utilización del instrumento es consistente y confiable.

Método de Test – Retest

Según Calderon (2013) menciona al método de Test-Retest “como el mismo proceso es aplicado a varios grupos de personas, luego de su medición. Si los resultados aplicados son positivos, el instrumento es considerado fiable [...]” (p.56).

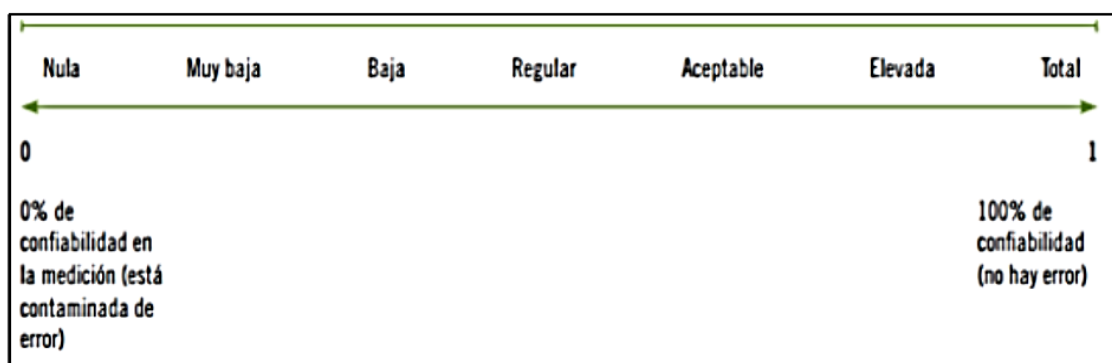
En esta investigación se empleó el método Test Retest y la técnica de coeficiente de correlación de Pearson.

Coeficiente de correlacion de Pearson

Según Nogale (2015) “varia de +1 a -1 aunque frecuentemente es transformado a porcentaje. El coeficiente no es algo estatico sino que cambia con el tiempo debido a los numerosos factores que afectan [...]” (p. 283).

En esta investigacion se preciso que en cada indicador la correlacion es elevada y aceptable.

Figura 11 Grado de confiabilidad



Fuente: Hernández, Fernández y Baptista (2014)

Confiabilidad

Para el indicador del nivel porcentual de pacientes atendidos en triaje

En la siguiente figura se efectúa el índice de correlación de Pearson que se atribuye al indicador del nivel porcentual de pacientes atendidos en triaje.

Tabla 7 Índice de correlación de Pearson para indicador Nivel porcentual de pacientes atendidos en triaje

		TEST_NPC A	RETEST_NPC CA
TEST_NPCA	Correlación de Pearson	1	,735**
	Sig. (bilateral)		.000
	N	20	20
RETEST_NPCA	Correlación de Pearson	,735**	1
	Sig. (bilateral)	.000	
	N	20	20

** . La correlación es significativa en el nivel 0,01 (bilateral).

Fuente: Elaboración propia

El índice de correlación de Pearson es de 0,735, por lo cual denota una elevada correlación entre el test y re-test efectuado al indicador de nivel porcentual de pacientes atendidos en triaje esto se evidencia en la tabla N°7, afirmando de esta manera que el instrumento de medición es confiable. (VER ANEXO 5)

Para el indicador del nivel percentil de cumplimiento

En la siguiente figura se efectúa el índice de correlación de Pearson que se atribuye al indicador del nivel percentil de cumplimiento.

Tabla 8 Índice de correlación de Pearson para indicador Nivel percentil de cumplimiento

		TEST_NPC C	RETEST_NPC C
TEST_NPC	Correlación de Pearson	1	,781**
	Sig. (bilateral)		.000

	N	20	20
RETEST_NPC	Correlación de Pearson	,781**	1
	Sig. (bilateral)	.000	
	N	20	20
**. La correlación es significativa en el nivel 0,01 (bilateral).			

Fuente: Elaboración propia

El índice de correlación de Pearson es de 0,781, por lo cual denota una admisible correlación entre el test y re-test efectuado al indicador de nivel percentil de cumplimiento esto se evidencia en la tabla N° 8, afirmando de esta manera que el instrumento de medición es fiable. (**VER ANEXO 5**).

2.5 Métodos de Análisis de Datos

Según Escudero (2014), define que “Una vez tabulados los datos, serán sometidos a diversas técnicas de análisis. Este análisis consiste en obtener, de los datos numéricos contenidos en las tablas, medidas estadísticas representativas de ellas, y en determinar la validez de estas medidas para el conjunto de la población objeto de estudio” (p.119).

En la presente investigación se utilizó el análisis de datos de forma cuantitativa, ya que es pre-experimental, para lo cual se obtendrán datos estadísticos que servirán a corroborar que la hipótesis alterna sea la correcta y la nula es desaprobado.

Hipótesis de investigación 1

a. Hipótesis Específica (HE1)

El sistema web utilizando el protocolo de Manchester aumenta el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en la Clínica Sanna San Borja.

b. Indicador 1: Nivel porcentual de pacientes atendidos en triaje

NPCAa: Nivel porcentual de pacientes atendidos en triaje antes de utilizar el sistema web utilizando el protocolo de Manchester.

NPCAd: Nivel porcentual de pacientes atendidos en triaje después de utilizar el sistema web utilizando el protocolo de Manchester.

c. Hipótesis Estadística 1:

Hipótesis Nula (H0):

El sistema web utilizando el protocolo de Manchester no aumenta el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en la Clínica Sanna San Borja.

$$H_0 : NPCA_a \geq NPCA_d$$

Se infiere que el indicador sin el sistema web utilizando el protocolo de Manchester es mejor que el indicador con el sistema web utilizando el protocolo de Manchester.

Hipótesis Alterna (HA):

El sistema web utilizando el protocolo de Manchester aumenta el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en la Clínica Sanna San Borja.

$$H_A : NPCA_a < NPCA_d$$

Se infiere que el indicador con el sistema web utilizando el protocolo de Manchester es mejor que el indicador sin el sistema web utilizando el protocolo de Manchester.

Hipótesis de investigación 2**a. Hipótesis Específica (HE2)**

El sistema web utilizando el protocolo de Manchester aumenta el nivel percentil de cumplimiento en el proceso de emergencia médica en la Clínica Sanna San Borja.

b. Indicador 2: Nivel percentil de cumplimiento

NPCa: Nivel percentil de cumplimiento antes de utilizar el sistema web utilizando el protocolo de Manchester.

NPCd: Nivel percentil de cumplimiento después de utilizar el sistema web utilizando el protocolo de Manchester.

c. Hipótesis Estadística 1:

Hipótesis Nula (H0):

El sistema web utilizando el protocolo de Manchester no aumenta el nivel percentil de cumplimiento en el proceso de emergencia médica en la Clínica Sanna San Borja.

$$H_0 : NPCa \geq NPCd$$

Se infiere que el indicador sin el sistema web utilizando el protocolo de Manchester es mejor que el indicador con el sistema web utilizando el protocolo de Manchester.

Hipótesis Alterna (HA):

El sistema web utilizando el protocolo de Manchester aumenta el nivel percentil de cumplimiento en el proceso de emergencia médica en la Clínica Sanna San Borja.

$$H_A : NPCa < NPCd$$

Se infiere que el indicador con el sistema web utilizando el protocolo de Manchester es mejor que el indicador sin el sistema web utilizando el protocolo de Manchester.

Nivel de significancia

El nivel de significancia utilizado fue $\alpha = 5\%$ (error), equivalente a 0.05, esto permitió realizar la comparación para que se tome la decisión de aceptar o rechazar la hipótesis.
Nivel de confiabilidad: $(1-\alpha) = 0.95$

Estadística de prueba

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{S_1^2}{N_1} + \frac{S_2^2}{N_2}}}$$

Dónde:

S1 = Varianza grupo Pre-Test

S2 = Varianza grupo Post-Test

= Media muestral Pre-Test

= Media muestral Post-Test

N = Número de muestra (Pre-Test y Post-Test)

Región de Rechazo

La región de rechazo es $t = t_x$

Donde t_x es tal que:

$P [T > T_x] = 0.05$, donde t_x = Valor Tabular

Luego Región de Rechazo: $T > T_x$

Cálculo de la Media

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Cálculo de la Varianza

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

Desviación Estándar

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

Dónde:

\bar{x} = Media

s^2 = Desviación Estándar

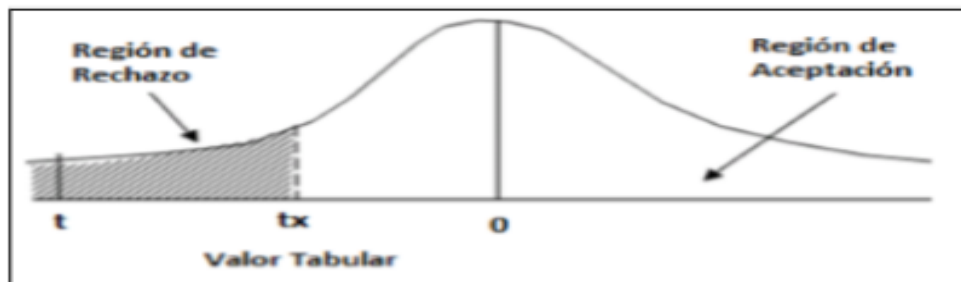
σ^2 = Calculo de varianza

x_i = Dato i que está entre (0, n)

\bar{x} = Promedio de los datos

n = Número de datos

Distribución T-Student



Distribución T-Student

Fuente: Hernández (2010)

2.6. Aspectos éticos

Se siguieron las directrices y normativas de la Universidad César Vallejo.

Los datos recogidos y procesados de la presente investigación no fueron adulterados, debido que estos datos están fundamentados en el instrumento aplicado en el pre test de estudio. La identidad de los que cooperaron en la investigación y los resultados conseguidos están resguardados y reservados.

La información utilizada solo se usó para fines académicos y sin fines comerciales u otros aspectos en donde los investigadores se comprometieron a salvaguardar la información brindada por la empresa.

Por último, los resultados de la investigación hasta la fecha no han sido adulteradas ni copiadas de otras investigaciones y se hizo un buen uso de la investigación en beneficio de otros.

III. RESULTADOS

3.1. Análisis Descriptivo

En el estudio se aplicó un Sistema Web para evaluar el nivel porcentual de pacientes atendidos en triaje y el nivel percentil de cumplimiento; para ello se aplicó un Pre-Test que permita conocer las condiciones iniciales del indicador; posteriormente se implementó el Sistema Web utilizando el protocolo de Manchester y nuevamente se registró el nivel porcentual de pacientes atendidos en triaje y el nivel percentil de cumplimiento. Los resultados descriptivos de estas medidas se observan en las tablas 9 y 10.

- **INDICADOR: Nivel porcentual de pacientes atendidos en Triaje**

Los resultados descriptivos del nivel porcentual de pacientes atendidos en triaje de estas medidas se observan en la Tabla 9.

Tabla 9 Medidas descriptivas del nivel porcentual de pacientes atendidos en triaje y después de implementar el Sistema Web utilizando el protocolo de Manchester

Estadísticos descriptivos

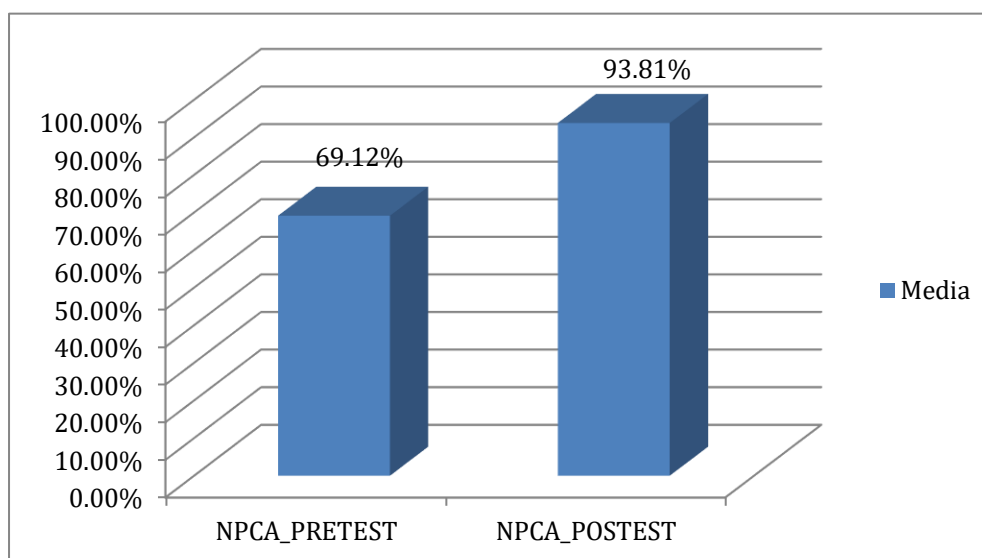
	N	Mínimo	Máximo	Media	Desviación estándar
NPCA_PRETEST	20	66.67	71.43	69.1190	1.41934
NPCA_POSTEST	20	92.31	95.00	93.8085	.60513
N válido (por lista)	20				

Fuente: Elaboración Propia

En el caso del nivel porcentual de pacientes atendidos en triaje, en el pre-test se obtuvo un valor de 69.12%, mientras que en el post-test fue de 93.80% tal como se aprecia en la Figura 12; esto indica una gran diferencia antes y después de la implementación del Sistema Web utilizando el protocolo de Manchester; así mismo, el porcentaje del nivel porcentual de pacientes atendidos en triaje mínima fue de 66.67% antes, y 92.31% (ver Tabla 9) después de la implementación del Sistema Web utilizando el protocolo de Manchester.

En cuanto a la dispersión del índice de calidad, en el pre-test se tuvo una variabilidad de 1.42%; sin embargo, en el post-test se tuvo un valor de 0.61%.

Figura 12 Porcentaje del nivel porcentual de pacientes atendidos en triaje antes y después de implementar el Sistema Web utilizando el protocolo de Manchester



Fuente: Elaboración Propia

- **INDICADOR: Nivel Percentil de Cumplimiento**

Los resultados descriptivos del nivel percentil de cumplimiento de estas medidas se observan en la Tabla 10.

Tabla 10 Medidas descriptivas del nivel percentil de cumplimiento antes y después de la implementación del Sistema Web utilizando el protocolo de Manchester

Estadísticos descriptivos

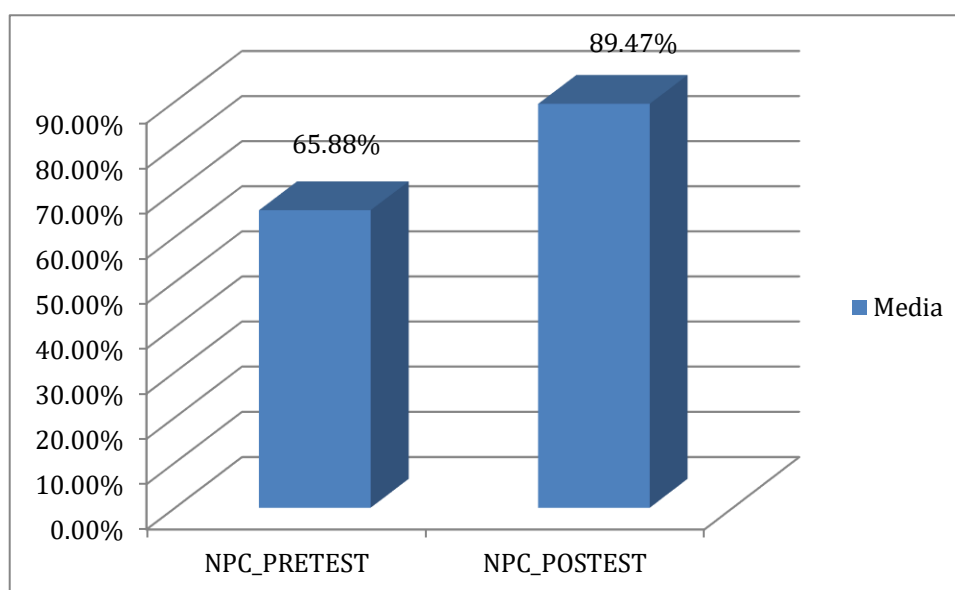
	N	Mínimo	Máximo	Media	Desviación estándar
NPC_PRETEST	20	61.90	68.18	65.8750	1.74313
NPC_POSTEST	20	81.25	95.00	89.4740	3.70186

N válido (por lista)	20				
----------------------	----	--	--	--	--

En el caso del nivel percentil de cumplimiento, en el pre-test se obtuvo un valor de 65.88%, mientras que en el post-test fue de 89.47% tal como se aprecia en la Figura 13; esto indica una gran diferencia antes y después de la implementación del Sistema Web utilizando el protocolo de Manchester; así mismo, el porcentaje del nivel percentil de cumplimiento mínima fue del 61.90% antes, y 81.25% (Ver tabla 10) después de la implementación del Sistema Web utilizando el protocolo de Manchester.

En cuanto a la dispersión del porcentaje del nivel percentil de cumplimiento, en el pre-test se tuvo una variabilidad de 1.74%; sin embargo, en el post-test se tuvo un valor de 3.70%.

Figura 13 Porcentaje del nivel percentil de cumplimiento antes y después de la implementar el Sistema Web utilizando el protocolo de Manchester



Fuente: Elaboración Propia

3.2. Análisis Inferencial

Prueba de Normalidad

Se procedió a realizar las pruebas de normalidad para los indicadores del nivel porcentual de pacientes atendidos en triaje y el nivel percentil de cumplimiento completos a través del método Shapiro-Wilk, debido a que el tamaño de nuestra muestra estratificada está conformado por 20 fichas de registro y es menor a 50, tal como lo indica Hernández, Fernández y Baptista (2014, p. 376). Dicha prueba se realizó introduciendo los datos de cada

indicador en el software estadístico SPSS 24.0, para un nivel de confiabilidad del 95%, bajo las siguientes condiciones:

Si:

Sig. < 0.05 adopta una distribución no normal.

Sig. \geq 0.05 adopta una distribución normal.

Dónde:

Sig.: P-valor o nivel crítico del contraste.

Los resultados fueron los siguientes:

- **INDICADOR: Nivel Porcentual de Pacientes Atendidos en Triage**

Con el objetivo de seleccionar la prueba de hipótesis; los datos fueron sometidos a la comprobación de su distribución, específicamente si los datos del nivel porcentual de pacientes atendidos en triaje contaban con distribución normal.

Tabla 11 Prueba de Normalidad del nivel porcentual de pacientes atendidos en triaje antes y después de la implementación del Sistema Web utilizando el protocolo de Manchester

Pruebas de normalidad

	Shapiro-Wilk		
	Estadístico	gl	Sig.
NPCA_PRETEST	.918	20	.090
NPCA_POSTEST	.919	20	.094

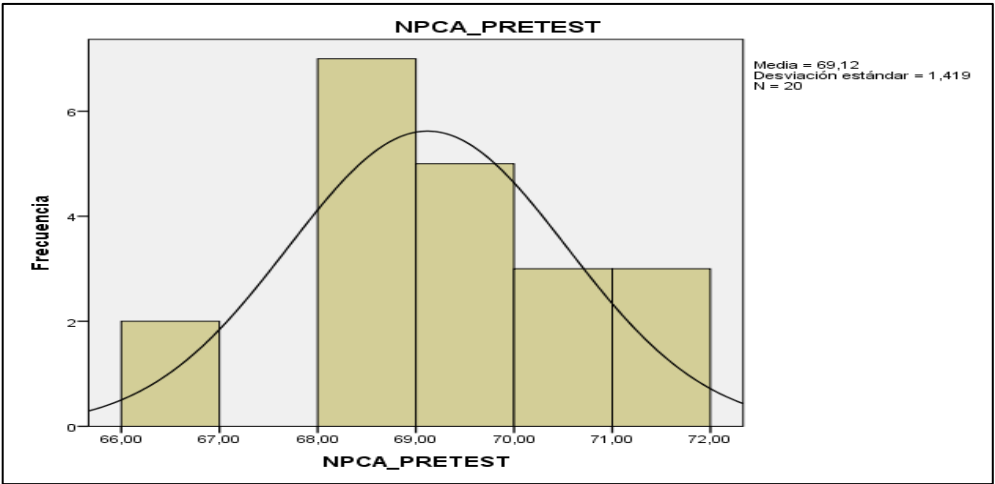
a. Corrección de significación de Lilliefors

Fuente: Elaboración Propia

Como se muestra en la Tabla 11 los resultados de la prueba indican que el Sig. Del nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en el Pre-Test fue de 0.090, cuyo valor es mayor que 0.05. Por lo tanto, el nivel porcentual de pacientes atendidos en triaje se distribuye normalmente. Los resultados de la prueba del Post-Test indican que el Sig. del nivel porcentual de pacientes atendidos en triaje fue de 0.094, cuyo valor es mayor que 0.05, por lo que

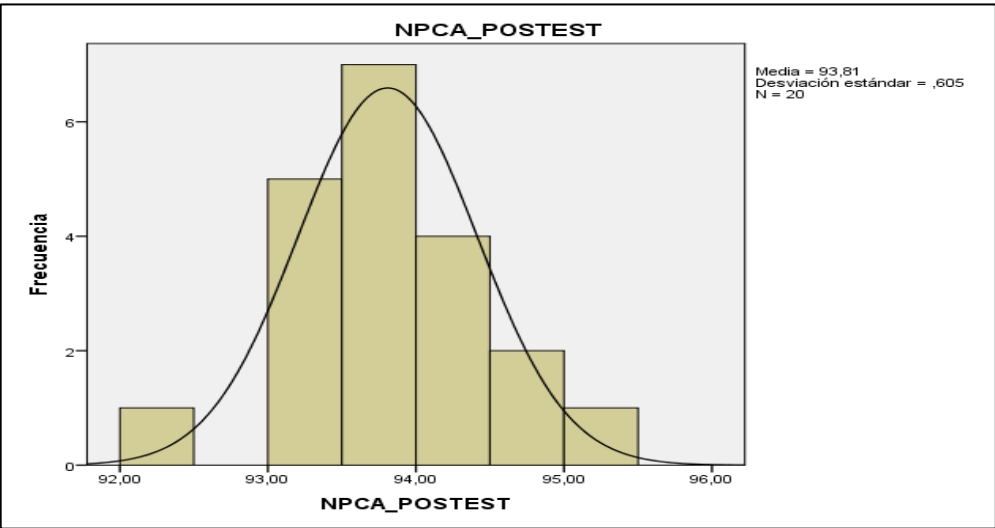
indica que el nivel porcentual de pacientes atendidos en triaje se distribuye normalmente. Lo que confirma la distribución normal de ambos datos de la muestra, se puede apreciar en las Figuras 14 y 15.

Figura 14 Prueba de Normalidad del nivel porcentual de pacientes atendidos en triaje antes de implementar el Sistema Web utilizando el protocolo de Manchester



Fuente: Elaboración Propia

Figura 15 Prueba de normalidad del nivel porcentual de pacientes atendidos en triaje de después de implementar el Sistema Web utilizando el protocolo de Manchester



Fuente: Elaboración Propia

- **INDICADOR: Nivel Percentil de Cumplimiento**

Con el objetivo de seleccionar la prueba de hipótesis; los datos fueron sometidos a la comprobación de su distribución, específicamente si el nivel percentil de cumplimiento contaba con distribución normal.

Tabla 12 Prueba de normalidad del nivel percentil de cumplimiento antes y después de implementado el Sistema Web utilizando el protocolo de Manchester

Pruebas de normalidad

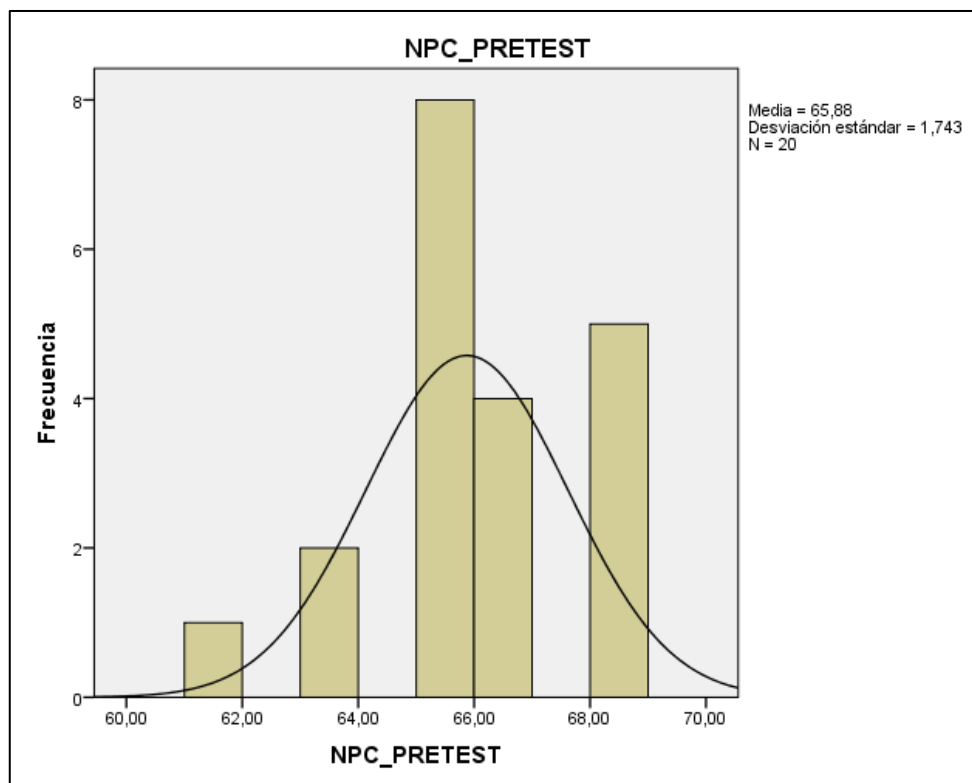
	Shapiro-Wilk		
	Estadístico	gl	Sig.
NPC_PRETEST	.913	20	.073
NPC_POSTEST	.910	20	.065

a. Corrección de significación de Lilliefors

Fuente: Elaboración Propia

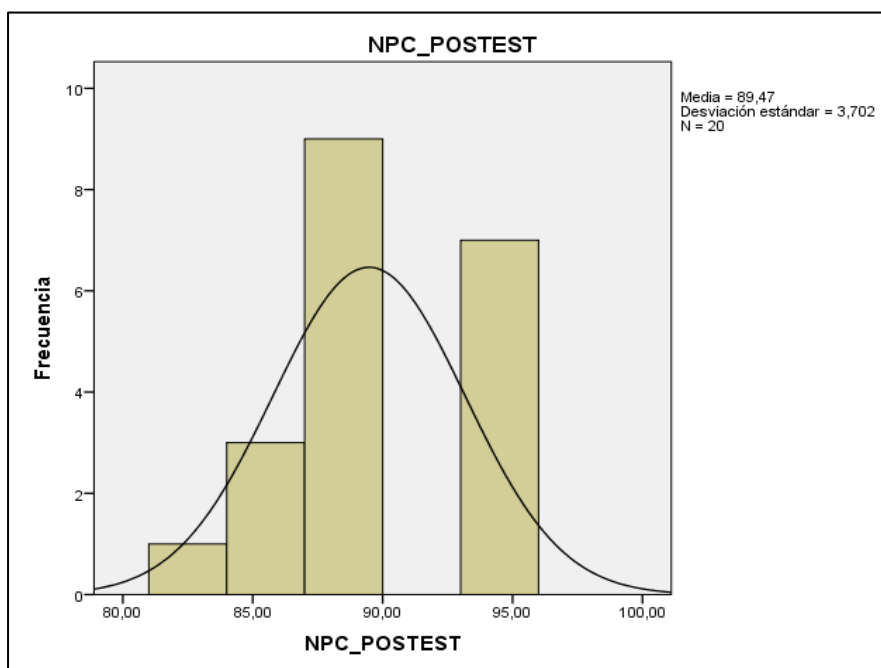
Como se muestra en la Tabla 12, los resultados de la prueba indican que el Sig. del nivel percentil de cumplimiento en el proceso de control de pedidos en el Pre-Test fue de 0.073, cuyo valor es mayor que 0.05, por lo que indica que el nivel percentil de cumplimiento se distribuye normalmente. Los resultados de la prueba del Post-Test indican que el Sig. de los Entregados completos fue de 0.065, cuyo valor es mayor que 0.05, por lo que indica que el nivel percentil de cumplimiento se distribuye normalmente. Lo que confirma la distribución normal de ambos datos de la muestra, lo cual se puede apreciar en las Figuras 16 y 17.

Figura 16 Prueba de normalidad del nivel percentil de cumplimiento antes de implementar el Sistema Web utilizando el protocolo de Manchester



Fuente: Elaboración Propia

Figura 17 Prueba de normalidad del nivel percentil de cumplimiento después de implementar el Sistema Web utilizando el protocolo de Manchester



Fuente: Elaboración Propia

3.2.Prueba de Hipótesis

Hipótesis de Investigación 1:

H1: El Sistema Web utilizando el protocolo de Manchester incrementa el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica de la clínica Sanna San Borja.

- **Indicador:** Nivel Porcentual de Pacientes Atendidos en Triaje

Hipótesis Estadísticas

Definiciones de Variables:

NPCAa: Nivel Porcentual de Pacientes Atendidos en Triaje antes de utilizar el Sistema Web utilizando el protocolo de Manchester.

NPCAd: Nivel Porcentual de Pacientes Atendidos después de utilizar el Sistema Web utilizando el protocolo de Manchester.

H0: El Sistema Web utilizando el protocolo de Manchester no incrementa el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica de la Clínica Sanna San Borja.

$$H_0: NPCAa \geq NPCAd$$

El indicador sin el Sistema Web utilizando el protocolo de Manchester es mejor que el indicador con el Sistema Web utilizando el protocolo de Manchester.

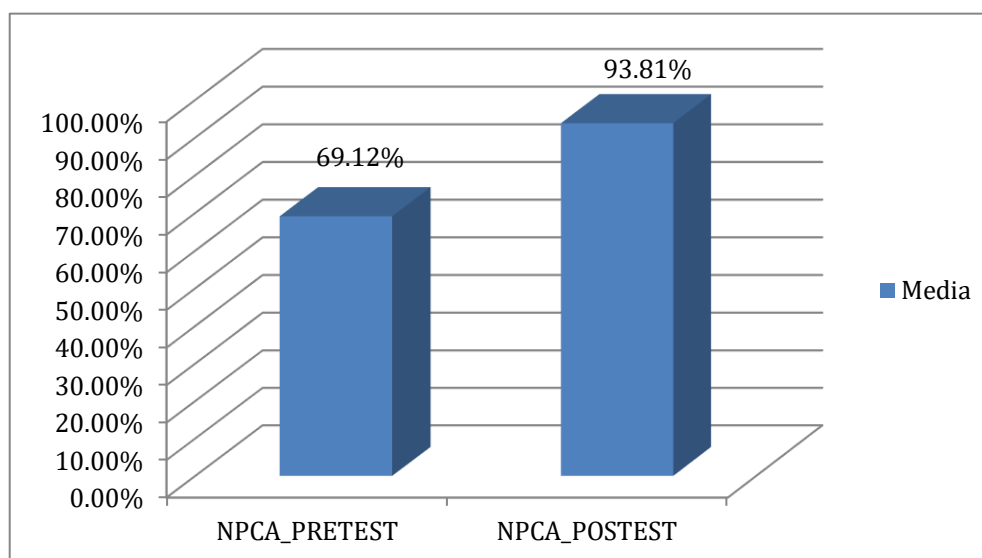
HA: El Sistema Web utilizando el protocolo de Manchester incrementa el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica de la Clínica Sanna San Borja

$$H_a: NPCAa < NPCAd$$

El indicador con el Sistema Web utilizando el protocolo de Manchester es mejor que el indicador sin el Sistema Web utilizando el protocolo de Manchester.

En la Figura 18, el nivel porcentual de pacientes atendidos en triaje (Pre Test), es de 69.12% y el Post-Test es 93.81%.

**Figura 18 Nivel Porcentual de Pacientes
Atendidos en Triage - Comparativa General**



Fuente: Elaboración Propia

Se concluye de la Figura 18 existe un incremento en el nivel porcentual de pacientes atendidos en triaje, el cual se puede verificar al comparar las medias respectivas, que asciende de 69.12% al valor de 93.81%.

En cuanto al resultado del contraste de hipótesis se aplicó la Prueba T-Student, debido a que los datos obtenidos durante la investigación (Pre-Test y Post-Test) se distribuyen normalmente. El valor de T contraste es de -99,136, el cual es claramente menor que - 1.7291 (Ver tabla 13).

**Tabla 13 Prueba de T-Student para el nivel porcentual de pacientes
atendidos en triaje en el proceso de emergencia médica antes y después de
implementar el Sistema Web utilizando el protocolo de Manchester**

	Media	Prueba de T-Student		
		T	gl	Sig. (bilateral)
NPCA_PRETEST	69,1190	- 99,136	19	0.000
NPCA_POSTEST	93,8085			

Fuente: Elaboración Propia

Entonces, se rechaza la hipótesis nula, aceptando la hipótesis alterna con un 95% de confianza. Además, el valor T obtenido, como se muestra en la Figura 19, se ubica en la zona de rechazo. Por lo tanto, El Sistema Web utilizando el protocolo de Manchester incrementa el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica de la clínica Sanna San Borja.

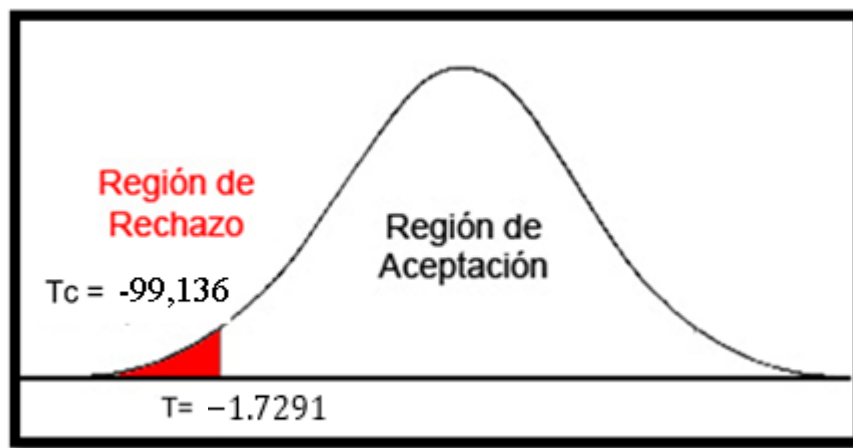
Aplicando la formula T Student:

$$T_c = \frac{x - u}{S/\sqrt{n}}$$

$$T_c = \frac{6911 - 9380}{1.1137/\sqrt{20}}$$

$$T_c = -99.136$$

Figura 19 Prueba T-Student – Nivel porcentual de pacientes atendidos en triaje



Fuente: Elaboración Propia

Hipótesis de Investigación 2:

H2: El Sistema Web utilizando el protocolo de Manchester incrementa el Nivel percentil de cumplimiento en el proceso de emergencia médica de la Clínica Sanna San Borja

- **Indicador:** Nivel Percentil de Cumplimiento

Hipótesis Estadísticas

Definiciones de Variables:

NPCa: Nivel percentil de cumplimiento antes de usar el Sistema Web utilizando el protocolo de Manchester.

NPCd: Nivel percentil de cumplimiento después de usar el Sistema Web utilizando el protocolo de Manchester.

H0: El Sistema Web utilizando el protocolo de Manchester no incrementa el nivel percentil de cumplimiento en el proceso de emergencia médica de la Clínica Sanna San Borja

$$H0 = NPCa \geq NPCd$$

El indicador sin el Sistema Web utilizando el protocolo de Manchester es mejor que el indicador con el Sistema Web utilizando el protocolo de Manchester.

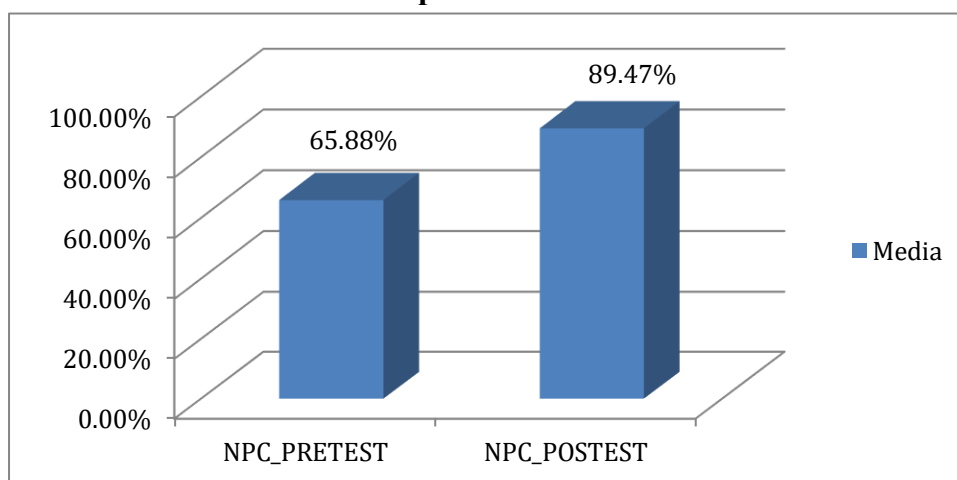
HA: El Sistema Web utilizando el protocolo de Manchester incrementa el nivel percentil de cumplimiento en el proceso de emergencia médica de la Clínica Sanna San Borja.

$$HA = NPCa < NPCd$$

El indicador con el Sistema Web utilizando el protocolo de Manchester es mejor que el indicador sin el Sistema Web utilizando el protocolo de Manchester.

En la Figura 20, Nivel percentil de cumplimiento (Pre Test), es de 65.88% y el Post-Test es de 89.47%.

Figura 20 Nivel percentil de cumplimiento - Comparativa General



Fuente: Elaboración Propia

Se concluye de la Figura 20 existe un incremento en el nivel percentil de cumplimiento, el cual se puede verificar al comparar las medias respectivas, que asciende de 65.88% al valor de 89.47%.

En cuanto al resultado del contraste de hipótesis se aplicó la Prueba T-Student, debido a que los datos obtenidos durante la investigación (Pre-Test y Post-Test) se distribuyen normalmente. El valor de T contraste es de -25,729, el cual es claramente menor que -1.7291 (Ver tabla 14).

Tabla 14 Prueba de T-Student para el Nivel percentil de cumplimiento en el proceso de emergencia médica antes y después de implementar el Sistema Web utilizando el protocolo de Manchester

	Media	Prueba de T-Student		
		T	gl	Sig. (bilateral)
NPC_PRETEST	65,8750	-25,729	19	,000
NPC_POSTEST	89,4740			

Fuente: Elaboración Propia

Entonces, se rechaza la hipótesis nula, aceptando la hipótesis alterna con un 95% de confianza. Además, el valor T obtenido, como se muestra en la Figura 21 se ubica en la zona de rechazo. Por lo tanto, El Sistema Web utilizando el protocolo de Manchester incrementa el nivel percentil de cumplimiento en el proceso de emergencia médica de la Clínica Sanna San Borja

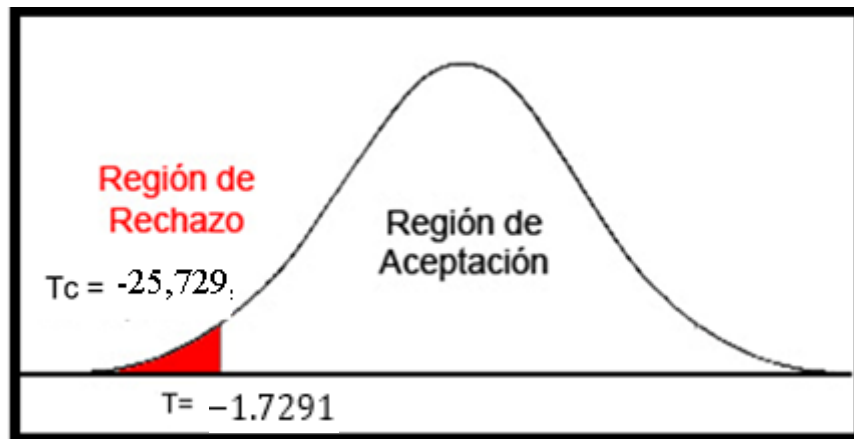
Aplicando la formula T Student:

$$Tc = \frac{x - u}{S/\sqrt{n}}$$

$$Tc = \frac{6587 - 8947}{0.9172/\sqrt{20}}$$

$$Tc = -25.729$$

Figura 21 Prueba T-Student – Nivel percentil de cumplimiento



Fuente: Elaboración Propia

IV. DISCUSIÓN

DISCUSIÓN

En primer lugar, con el propósito de validar la hipótesis planteada en este trabajo de investigación: “Sistema Web utilizando el protocolo de Manchester para el proceso de emergencia médica en la Clínica Sanna San Borja” y sustentar el cumplimiento del objetivo general: “Determinar la influencia de un sistema web utilizando el protocolo de Manchester en el proceso de emergencia médica en la Clínica Sanna Borja”. Se demuestra que según los resultados conseguidos en la investigación en base a los hallazgos encontrados para los indicadores: Nivel porcentual de pacientes atendidos en triaje y el Nivel percentil de cumplimiento, se realiza una comparación con otras investigaciones.

Los resultados conseguidos en la investigación, nos indican la mejora del indicador con la implementación del Sistema Web utilizando el protocolo de Manchester, incrementó el nivel porcentual de pacientes atendidos en triaje de 69,12% a 93,81%, demostrando un aumento promedio de 24,69%. Así mismo, Duarte (2016), en su estudio “Sistema web para la administración de atenciones médicas y monitoreo de la unidad de triaje en el Hospital Cayetano Heredia”, el cual el índice del nivel de pacientes atendidos en triaje es de un 77% con lo cual se concluyó que la presente investigación obtuvo una mejora de 88% al implementar el sistema web, en donde se evidenció un incremento de 11%.

Por otro lado, se evidencia que el Sistema Web utilizando el protocolo de Manchester incremento el nivel percentil de cumplimiento de 65,88% a 89,47 lo que equivale a un aumento promedio de 23,60%. Así mismo, Chuquilin Llanos y Vásquez Cabrera (2018), en su estudio “Implementación de un sistema informático para la gestión de atenciones a los pacientes del puesto de salud Agocucho del distrito de Cajamarca, 2016”, el cual el índice de atenciones del nivel percentil de cumplimiento es de un 60% con lo cual se concluyó que la presente investigación obtuvo una mejora de 72% con respecto al antecedente presentado; por lo que, se afirma que el indicador de nivel percentil de cumplimiento incrementaron con la implementación del Sistema web un promedio de 12%

Los resultados conseguidos en la investigación, nos indican la mejora del indicador con la implementación del Sistema Web utilizando el protocolo de Manchester, incrementó el nivel porcentual de pacientes atendidos en triaje de 69,12% a 93,81%, demostrando un aumento promedio de 24,69%. Así mismo, Torque y Rodrigues (2013), en su estudio “Desenvolvimento de um Sistema de Informação para Triagem com Classificação de risco nos Serviços de Urgência e Emergência”, el cual el índice del cumplimiento de pacientes atendidos en triaje es de un 68% con lo cual se concluyó que la presente investigación obtuvo una mejora de 90% al implementar el sistema web, en donde se evidenció un incremento de 22%.

Por otro lado, se evidencia que el Sistema Web utilizando el protocolo de Manchester incremento el nivel percentil de cumplimiento de 65,88% a 89,47 demostrando un aumento promedio de 23,60%. Así mismo, Monroy y Silva (2018), en su investigación tesis “Desarrollo e implementación de un sistema web para el Hospital León Becerra de Guayaquil enfocado en el registro de emergencia utilizando el lenguaje de programación PHP y el Framework Laravel”, el cual el índice de atenciones del nivel percentil de cumplimiento es de un 62% con lo cual se concluyó que la presente investigación obtuvo una mejora de 80% con respecto al antecedente presentado; por lo que, se afirma que el cumplimiento de atenciones de los pacientes en emergencia incrementaron con la implementación del Sistema web un promedio de 18%.

Estos resultados arrojados por la presente investigación nos permiten afirmar que el uso de la tecnología nos posibilita determinar diversos factores que impactan en los procesos de las organizaciones; siendo así que para esta investigación los indicadores evaluados en el proceso de emergencia médica, confirman que el Sistema Web utilizando el protocolo de Manchester para el proceso de emergencia médica de la Clínica Sanna San Borja incrementa el nivel porcentual de pacientes atendidos en triaje en un 24,69% y de la misma forma el nivel percentil de cumplimiento en un 23,60%, de los resultados conseguidos se infiere que el Sistema Web utilizando el protocolo de Manchester mejora del proceso de emergencia médica.

V. CONCLUSIONES

CONCLUSIONES

En conclusión, el Sistema Web utilizando el protocolo de Manchester mejora el proceso de emergencia médica de la Clínica Sanna San Borja, mediante el incremento del nivel porcentual de pacientes atendidos en triaje y el nivel percentil de cumplimiento, permitiendo no sólo alcanzar los objetivos de la investigación, sino que validar las hipótesis planteadas.

Para el Indicador Nivel Porcentual de Pacientes Atendidos en Triage nos arrojó que con la implementación del Sistema Web utilizando el protocolo de Manchester el indicador incrementó en un 24.69%. Por consiguiente, se rechazó la hipótesis nula y se afirma que el Sistema Web utilizando el protocolo de Manchester incrementa el Nivel Porcentual de Pacientes Atendidos en Triage en el Proceso de Emergencia Médica.

El Indicador Nivel Percentil de Cumplimiento nos arrojó que con la implementación del Sistema Web utilizando el protocolo de Manchester el indicador incrementó en un 23.60%. Por consiguiente, se rechazó la hipótesis nula y se afirma que el Sistema Web utilizando el protocolo de Manchester incrementa el Nivel Percentil de Cumplimiento en el Proceso de Emergencia Médica.

VI. RECOMENDACIONES

RECOMENDACIONES

Se recomienda el planteamiento de futuras investigaciones y/o tener una propuesta que permita seguir con la mejora del proceso de emergencia médica, y los demás procesos relacionados a la misma, que con conjunto puedan permitir una mejoría en los procesos en general de la empresa y generar valor agregado a la misma.

Así mismo, cada indicador analizado en la presente tesis, logró demostrar en base los resultados que son recomendables para tener en cuenta en futuras investigaciones, siendo así que, el indicador nivel porcentual de pacientes atendidos en triaje referencia a la mejoría con respecto a incremento del escalamiento y el indicador nivel percentil de cumplimiento que referencia al Cierre de las atenciones, en ambos casos se sugiere tomar en cuenta en base a las métricas y resultados obtenidos.

VII. REFERENCIAS

ACOSTA, Dionisio. Metodologías de análisis para la investigación de mercados [en línea]. 5° ed. España: Editorial Elearning S.L., 2015. [fecha de consulta 1 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?id=2GJWDwAAQBAJ>

ISBN 978-84-16492-32-9

ALONSO, Marina, ALONSO, Sonia y CABRAL, Gala. Evaluación de la Calidad del Sistema de Triage de HUBU [en línea]. España: Universidad de Burgos, 2016 [fecha de consulta 30 de abril de 2019]. Disponible en: <http://hdl.handle.net/10259/4069>

ISSN: 0124-8146

ALVARENGA Molina, [et al.] . Sistema de Información para el área de Hospitalización en el Hospital Nacional Rosales. [En línea]. San Salvador: Universidad de El Salvador, 2015 [fecha de consulta: 29 de noviembre de 2019]. Disponible en: <http://ri.ues.edu.sv/535/1/10137168.pdf>

Amthauer y Chollopetz. Sistema de Triage de Manchester: principais fluxogramas, discriminadores e desfechos dos atendimentos de uma emergência pediátrica [en línea]. Agosto 2016, vol. 24. [Fecha de consulta: 24 de abril de 2019]. Disponible en: <http://dx.doi.org/10.1590/1518-8345.1078.2779>

ISSN 1518-8345

ATAHUICHI Mamani, Glaucia. Sistema web de control de pedidos y ventas caso: empresa ITSEVEN soluciones informáticas integrales. Tesis (titulación en informática). La Paz: Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, 2014. 26 pp.

Báez, Patrick. Teoría de sistemas web y su desarrollo. 1a. ed. México, Parainfo, 2014. p.5., ISBN: 120234586-1

BELTRÁN, Marta y SEVILLANO, Fernando. Cloud Computing, tecnología y negocio [en línea]. 2º ed. España: Editorial Paraninfo, 2013 [fecha de consulta 2 de mayo de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=8428335141>
ISBN:8428335141

BIBIANO, Carlos. Manual de Urgencias [en línea]. 3º ed. Madrid: Editorial Grupo Saned, 2018 [fecha de consulta 30 de abril de 2019]. Disponible en: <https://www.urgenciasyemergen.com/wp.../10/Manual-de-urgencias-3ed-Bibiano.pdf>
ISBN 978-84-978-84-16831-98-2

BONETTO, Noelia. Prisión trabajo entregado. Como priorizar el superior interés del niño [en línea]. 2º ed. Editorial GRIN Verlag, 2018. [fecha de consulta 12 de mayo de 2019] Disponible en: <https://sites.google.com/site/kubarseaduc/home/prision-domicil-ry1dumi2q2r8f>
ISBN 3668680590

BRIONES Mariño y PAUL Alex. Implementación de un Sistema Web para la Gestión Administrativa en el Centro de Terapia Física y Rehabilitación de la Clínica Good Hope, Miraflores 2018. [En línea]. Lima: UPA, 2018 [fecha de consulta: 29 de noviembre de 2019]. Disponible en: <http://repositorio.ulasamericas.edu.pe/handle/upa/376>

CABALLERO, Verónica y ZAPATA, Rosa. Educación y salud en grupos vulnerables [en línea]. 2º ed. España: Universidad de Almería, 2019. [fecha de consulta 15 de mayo de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=841726129X>
ISBN 841726129X

CARDADOR, Luis. Implantación de aplicaciones web en entornos internet, intranet y extranet [en línea]. 2º ed. Málaga: Editorial IC, 2014. [fecha de consulta 1 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=8416433097>
ISBN 8416433097

Cárdenas, Antonio y Roca, Juan. Tratado de medicina intensiva acceso web [en línea]. 2° ed. Barcelona, España. 2016 [Fecha de consulta 26 de abril de 2019].
<https://books.google.com.pe/books?isbn=8491130853>

ISBN 8491130853

Comparación de metodologías en aplicaciones web por Molina [et al]. 3C Tecnología. [en línea]. Marzo 2018, no 1. [fecha de consulta 9 de mayo de 2019]. Disponible en:
<https://www.3ciencias.com/articulos/articulo/comparacion-metodologias-aplicaciones-web/>

Curso de Ingeniería de Software por Ramos [et al]. España: Editorial IT Campus Academy, 2017. 211 pp.

ISBN 1544132530

CRUZ, Cinthia, OLIVARES, Socorro y GONZÁLES, Martin. Metodología de la investigación [en línea]. 2° ed. México: Editorial Patria, 2014. [fecha de consulta 1 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=6074388768>

ISBN 6074388768

CHUQUILIN Llanos y VÁSQUEZ Cabrera. Implementación de un Sistema Informático para la gestión de atenciones a los pacientes del puesto de salud Agocucho del distrito de Cajamarca, 2016. [En línea]. Cajamarca: UPAGU, 2018 [fecha de consulta: 29 de noviembre de 2019]. Disponible en: <http://repositorio.upagu.edu.pe/handle/UPAGU/663>

DÍAZ, Ana, ÁLVAREZ, Jorge y PARDILOS, Luis. Atención sanitaria a multiples victimas [en línea]. 2° ed. España: Editorial Ideaspropias, 2014. [fecha de consulta 15 de mayo de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=849839483X>

ISBN 849839483X

DIEZ, Isabel, MARTIN, Gregorio y MONTORO, María. Economía de la empresa [en línea]. 3° ed. Madrid: Editorial Paraninfo, 2014. [fecha de consulta 10 de mayo de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=8497328884>

ISBN 8497328884

DUEÑAS, Juan. UF2119 – Planificación de la investigación de mercados [en línea]. 6° ed. España: Editorial ELearning, 2015. [fecha de consulta 15 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?id=NGNWDwAAQBAJ>

ISBN 978-84-16557-19-6

Enfermero/a. Servicio Murciano de Salud. Diplomado Sanitario no especialista [en línea]. 5° ed. Madrid: Editorial CEP, 2017. [fecha de consulta 1 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=846818540X>

ISBN 846818540X

ESCUDERO, María. Marketing en la actividad comercial [en línea]. 2° ed. España: Editex, 2014. [fecha de consulta 01 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=8490036209>

ISBN 8490036209

Factores asociados al uso del servicio de urgencias en el Hospital San Vicente de Arauca entre el período septiembre de 2013 y agosto de 2014 por Dwyver Medina [et al]. Acta odontológica no.1 [Fecha de consulta: 27 de abril de 2019] Disponible en: <https://revistas.unal.edu.co/index.php/actaodontocol/article/view/70356>

ISSN 2027-7822

GRACIA, Ruben. Aprende a modelar aplicaciones con UML [en línea]. 2° ed. España: Editorial IT Campus Academy, 2016. [fecha de consulta: 12 de mayo de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=1523498536>
ISBN 1523498536

HAMUI, Liz, PAULO, Alfredo y HERNÁNDEZ, Isaías. La comunicación dialógica como competencia médica esencial [en línea]. México: Editorial El Manual Moderno, 2018 [fecha de consulta 30 de abril de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=6074486786>

ISBN 6074486786

HÉRNANDEZ, Roberto, FÉRNANDEZ, Carlos y BAPTISTA, Pilar. Metodología de la investigación. 5° ed. México: Mc Graw Hill, 2014. 322 pp.

ISBN 970-10-5753-8

Huamán, Pamela, Mosquera, Kevin, Cieza Javier. Supervivencia de pacientes hospitalizados en la emergencia de un hospital general nivel III de Lima-Perú. [en línea] Perú: Acta med.2018, no. 1 [Fecha de consulta: 27 de abril de 2019]. Disponible en: www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1728.

ISSN 1728-5917

HUAMAN Zamudio y ANGEL Jose. Implementación de un sistema web de historial Clínico para la mejora de los procesos de historias clínicas del centro de Salud de la Provincia de Cañete. [En línea]. Lima: UPA, 2017 [fecha de consulta: 29 de noviembre de 2019]. Disponible en: <http://repositorio.ulasamericas.edu.pe/handle/upa/531>

JIMENEZ, Julian. Manual de protocolos y actuación en urgencias. 4° ed. Toledo: España, 2014. pp 10.

ISBN 978-84-96835-57-3

La empresa en España por AGUILERA Jaime [et al]. España: Editorial Grupo Planeta, 2017, 10 pp.

ISBN 8423426548

LERMA, Héctor. Metodología de la investigación: Propuesta, anteproyecto y proyecto [en línea]. 23° ed. Bogotá: Editorial Andrés Delgado, 2016 [fecha de consulta 1 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=958771346X>

ISBN 958771346X

MARTINEZ, José. Investigación comercial [en línea]. 2° ed. Madrid: Paraninfo, 2015. [fecha de consulta 15 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=8428337136>

ISBN 8428337136

Medina, F. Metodología uwe uml (uml-based web engineering) [en línea]. Mexico: KUPDF, 2016 [fecha de consulta 09 de mayo de 2019]. Disponible en:

https://kupdf.net/download/metodologia-uwe-uml_5cd3c8fbe2b6f5221fe2dd8c_pdf#

ISBN: 2314-2642

Mendoza, Carmen y Quispe, Lin. Eficacia de las intervenciones de enfermería en el manejo del sistema de triaje Manchester para mejorar el flujo de atención de pacientes en el servicio de emergencia. [En línea]. Lima: WIENER, 2018 [fecha de consulta: 27 de abril de 2019]. Disponible en: repositorio.uwiener.edu.pe/handle/123456789/2510

ISSN 2308-0531.

Mesquita, Mirta, Pavlicich, Viviana y Luaces, Carlos. El sistema español de triaje en la evaluación de los neonatos en las urgencias pediátricas [en línea]. Chile Pediatría 2016, no 1. [Fecha de consulta 30 de abril de 2019]. Disponible en:

<http://revistachilenadepediatria.cl/index.php/rchped/article/viewFile/216/62>

ISSN: 0370-4106

MUÑOZ, Carlos. Metodología de la investigación [en línea]. 3° ed. Mexico: Editorial Oxford University, 2015. [fecha de consulta 1 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=6074265429>

ISBN 6074265429

MUÑOZ, Yanier [et al]. Sistema informático para la gestión de la información hospitalaria del infarto agudo de miocardio (RHIMA). Sociedad Cubana de Cardiología [en línea]. CorSalud 2015 Jul-Sep;7(3): 181-186 pp. [Fecha de consulta: 19 de abril de 019]. Disponible en: <http://www.revcorsalud.sld.cu/index.php/cors/article/view/56/93>
ISSN 2078-7170.

NAMAKFOROOSH, Mohammad. Metodología de la investigación. 2º ed. México: Limusa, 2013. 103 pp.
ISBN 978-968-18-5517-8

NIEVES, Citlali, UCÁN, Juan y MENÉNDEZ, Víctor. UWE en sistema de recomendación de objetos de aprendizaje. Aplicando ingeniería web: un método en caso de estudio. Latinoamericana Ingeniería [en línea]. México 2014, n.o 3. [fecha de publicación 12 de mayo de 2019]. Disponible en: revistas.unla.edu.ar/software/article/download/160/139
ISSN 2314-2642

PASIES, Laura. Comparativa entre los sistemas MTS Y SET-MAT. [En línea]. Castellón: Universitat Jaume, 2016 [fecha de consulta: 29 de noviembre de 2019]. Disponible en: http://repositori.uji.es/xmlui/bitstream/handle/10234/161456/TFG_2015_pasiesL.pdf?sequence=1&isAllowed=y

Primeros auxilios en urgencias y emergencias médicas por Leonardo Serrano [et al.]. Ecuador: editorial de la UC, 2015. 12 pp.
ISBN 978-9978-14-312-4

RAMOS, Alicia y RAMOS, María. Aplicaciones Web [en línea]. 2º ed. España: Editorial Paraninfo, 2014 [fecha de consulta 2 de mayo de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=8428398755> ISBN 8428398755
ISSN: 2254-4143

Relación entre los niveles de prioridad del triaje y tiempo de espera en la atención de pacientes de emergencia – ESSALUD, Lima-Perú 2016 por Rosita Osorio Yalán [et al]. UPCH [en línea]. 2016, no 4. [Fecha de consulta: 24 de abril de 2019]. Disponible en: repositorio.upch.edu.pe/handle/upch/641

Ricardo, Alvarez y Marta, Andromaco. Demora en tiempos de respuestas a códigos amarillos en un servicio de emergencias médicas de la ciudad de Córdoba [en línea]. n° [Fecha de consulta: 24 de abril de 2019]. Disponible en: <http://dx.doi.org/10.31052/1853.1180.v20.n3.14541>
ISSN 1852-9429.

SÁNCHEZ, Roberto. Gestión y psicología en empresas y organizaciones [en línea]. 2° ed. Madrid: Editorial ESIC, 2014. [fecha de consulta 10 de mayo de 2019] Disponible en: <https://books.google.com.pe/books?isbn=847356989X>

ISBN 847356989X

Saturación en los servicios de urgencias: análisis de cuatro hospitales de Medellín y simulación de estrategias 2018 por Restrepo [et al]. Gerencia y políticas de salud [en línea]. Junio 2018, no 34. [Fecha de consulta 30 de abril de 2019]. Disponible en: <https://doi.org/10.11144/Javeriana.rgps17-34.ssua>

ISSN: 1657-7027

Sistema web para la administración de atenciones médicas y monitoreo de la unidad de triaje en el hospital nacional Cayetano Heredia [en línea]. Perú: interfaces 2016 [fecha de consulta: 24 de abril de 2019]. Disponible en: <https://revistas.ulima.edu.pe/index.php/Interfases/article/view/1244/1204>

ISSN 1993-4912.

SOLER, Jorge y RAURICH, Jordi. Comunicación y atención primaria para alumnos de medicina [en línea]. 2° ed. España: Universidad de Lleida, 2015. [fecha de consulta 15 de mayo de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=8484097366>

ISBN 8484097366

SUCESO. Compendio de Normas del Seguro Social de Accidentes del Trabajo y Enfermedades Profesionales. Santiago de Chile, 2018. 12 pp.

TOBAR, Eduardo, RETAMAL, Andrea y GARRIDO, Nicolás. Elementos de gestión para un Servicio de Urgencia de un hospital universitario. Departamento de Medicina Interna Hospital Clínico Universidad de Chile [en línea]. Rev Hosp Clín Univ Chile 2014; 25: pp.189-200. [Fecha de consulta: 18 de abril de 2019]. Disponible en: https://www.redclinica.cl/Portals/0/Users/014/14/14/elementos_gestion_servicio_urgencia.pdf

TORQUE Alfredo y RODRIGUES Francis. Desenvolvimento de um Sistema de Informação para Triagem com Classificação de risco nos Serviços de Urgência e Emergência. [En línea]. Brasil: Universidade do sul de Santa Catarina, 2013 [fecha de consulta: 29 de noviembre de 2019]. Disponible en: <http://www.riuni.unisul.br/handle/12345/3253>

UF0683 - Traslado del paciente al centro sanitario [en línea] por Garcia Ruth[et al.]. 2º ed. España: Editorial Parainfo, 2018. [fecha de consulta 30 de abril de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=842839797X>

ISBN 842839797X

Un caso de estudio sobre la aplicación de UWE para la generación de sistemas web por Damián Rotta [et al.]. JAIO [en línea]. Septiembre 2016, no 46. [fecha de consulta 10 de mayo de 2019]. Disponible en:

<http://www.clei2017-46jaiio.sadio.org.ar/node/88>

ISSN 2451-7615

Utilización y positividad de exámenes complementarios de laboratorio en el hospital provincial de Las Tunas por Hernández [et al.]. Infomed [en línea]. Octubre 2017, no 5. [Fecha de consulta 30 de abril de 2019]. Disponible en:

http://revzoiomarinello.sld.cu/index.php/zmv/article/view/1164/pdf_432

ISSN 1029-3027

VASQUEZ-ALVA, Rolando [et al.]. 2016. Sobredemanda de atención médica en el servicio de emergencia de adultos de un hospital terciario, Lima, Perú. An. Fac. med. [en línea]. An Fac med. 2016;77(4): 379-385 pp. [Fecha de consulta: 18 de abril de 2019]. Disponible en: http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1025-55832016000400010&lng=es&nrm=iso.

ISSN 1025-5583.

VÉLEZ, Carolina. Dirección de la actividad empresarial de pequeños negocios o microempresas [en línea]. 5° ed. España: Editorial Elearning, 2017. [fecha de consulta 10 de mayo de 2019]. Disponible en: <https://books.google.com.pe/books?id=hV5WDwAAQBAJ>

ISBN 978-84-16102-84-6

VIGIL, Paulino. Metodología de la investigación clínica [en línea]. 2° ed. Berlin: XinXii, 2018. [fecha de consulta 01 de junio de 2019]. Disponible en: <https://books.google.com.pe/books?isbn=3962463577>

ISBN 3962463577

VIII. ANEXOS

Anexo 1: Matriz de Consistencia

Problemas	Objetivos	Hipótesis	Variable	Dimensiones	Indicadores	Métodos
Principal: ¿De qué manera influye un sistema web utilizando el protocolo de Manchester en el proceso de emergencia médica en la Clínica Sanna San Borja?	General: Determinar la influencia del sistema web utilizando el protocolo de Manchester en el proceso de emergencia médica en la Clínica Sanna San Borja.	General: El sistema web utilizando el protocolo de Manchester mejora el proceso de emergencia médica en la Clínica Sanna San Borja.	Independiente: X1: Sistema web utilizando el protocolo Manchester			Método de investigación: Hipotético deductivo Tipo de investigación: Explicativa, Experimental y Aplicada Diseño de estudio: Pre-experimental Método estadístico: T-Student
Secundarios: P1: ¿Cómo influye un sistema web utilizando el protocolo de Manchester en el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en la Clínica Sanna San Borja?	Específicos: O1: Determinar la influencia del sistema web utilizando el protocolo de Manchester en el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en la Clínica Sanna San Borja.	Específicos: H1: El sistema web utilizando el protocolo de Manchester aumenta el nivel porcentual de pacientes atendidos en triaje en el proceso de emergencia médica en la Clínica Sanna San Borja.	Dependiente Y1: Proceso de emergencia médica	Triaje	Nivel porcentual de pacientes atendidos en triaje	Población: 1493 registros de triaje agrupados en 20 reportes para el nivel porcentual de pacientes atendidos en triaje y 1211 registros de triaje agrupados en 20 reportes del nivel percentil de cumplimiento. Muestra: 306 registros de triaje lunes a viernes de un mes estratificado en 20 reportes para el nivel porcentual de pacientes atendidos en triaje y 292 registros de triaje lunes a viernes de un mes estratificado en 20 reportes para el nivel percentil de cumplimiento Muestreo: Probabilístico Técnica e instrumento de recolección de datos: Técnica: Fichaje Instrumento: Ficha de registro
P2: ¿Cómo influye un sistema web utilizando el protocolo de Manchester en el nivel percentil de cumplimiento en el proceso de emergencia médica en la Clínica Sanna San Borja?	O2: Determinar la influencia del sistema web utilizando el protocolo de Manchester en el nivel percentil de cumplimiento en el proceso de emergencia médica en la Clínica Sanna San Borja.	H2: El sistema web utilizando el protocolo de Manchester aumenta nivel percentil de cumplimiento en el proceso de emergencia médica en la Clínica Sanna San Borja.			Nivel percentil de cumplimiento	

Fuente: Elaboración Propia

**Anexo 2: Ficha Técnica instrumentos de recolección de datos PRE-TEST y
POS - TEST**

Autores		Oscar Garcia Del Aguila Lincoln King Panizo Gonzales
Nombre de instrumento		Ficha de Registro
Lugar		Clínica Sanna San Borja
Fecha de Aplicación		03 de abril del 2019
Objetivos		Determinar la influencia del sistema web utilizando el protocolo de Manchester en el proceso de emergencia médica en la Clínica Sanna San Borja.
Tiempo de duración		20 días (lunes a viernes)
Elección de técnica de instrumento		
Variable	Técnica	Instrumento
Variable dependiente	Fichaje	Ficha de registro
Proceso de emergencia médica		
Variable independiente		
Sistema web utilizando el protocolo de Manchester		

Fuente: Elaboración Propia

Autores		Oscar Garcia Del Aguila Lincoln King Panizo Gonzales
Nombre de instrumento		Ficha de Registro
Lugar		Clínica Sanna San Borja
Fecha de Aplicación		04 de noviembre de 2019
Objetivos		Determinar la influencia del sistema web utilizando el protocolo de Manchester en el proceso de emergencia médica en la Clínica Sanna San Borja.
Tiempo de duración		20 días (lunes a viernes)
Elección de técnica de instrumento		
Variable	Técnica	Instrumento
Variable dependiente	Fichaje	Ficha de registro
Proceso de emergencia médica		
Variable independiente		
Sistema web utilizando el protocolo de Manchester		

Fuente: Elaboración Propia

Anexo 3: Fichas de registro

Nivel Porcentual de pacientes atendidos en triaje (Test)

FICHA DE REGISTRO						
Investigadores	Oscar García Del Aguila y. Lincoln King Panizo Gonzales		Tipo de prueba	Test		
Institución investigada	Clinica San Borja – Sanna					
Dirección	Av. Guardia Civil 337					
Motivo de Investigación	Nivel Porcentual de pacientes atendidos en triaje					
Fecha Inicio	01 de febrero		Fecha Fin	28 de febrero		
Variable	Indicador	Medida	Formula			
Proceso de emergencia médica	Nivel Porcentual de pacientes atendidos en triaje	Porcentaje	<div style="border: 1px solid black; padding: 2px;"> $\frac{CPC}{CTP} \times 100\%$ </div>			
Ítem	Fecha	Internista	Turno	Cantidad de pacientes admitidos (CTP)	Cantidad de pacientes atendidos en triaje (CPC)	Nivel Porcentual de pacientes atendidos en triaje (NPCA)
1	01-02-2019	352721	Diurno	110	77	70.00%
2	04-02-2019	365462	Diurno	107	73	68.22%
3	05-02-2019	483696	Diurno	108	75	69.44%
4	06-02-2019	352721	Diurno	106	74	69.81%
5	07-02-2019	365462	Diurno	104	71	68.27%
6	08-02-2019	365462	Diurno	114	77	67.54%
7	11-02-2019	352721	Diurno	112	76	67.86%
8	12-02-2019	365462	Diurno	114	77	67.54%
9	13-02-2019	352271	Diurno	107	74	69.16%
10	14-02-2019	409609	Diurno	107	75	70.09%
11	15-02-2019	352721	Diurno	109	77	70.64%
12	18-02-2019	476274	Diurno	103	70	67.96%
13	19-02-2019	352721	Diurno	110	74	67.27%
14	20-02-2019	396951	Diurno	103	70	67.96%
15	21-02-2019	352721	Diurno	105	72	68.57%
16	22-02-2019	365462	Diurno	115	79	68.70%
17	25-02-2019	421869	Diurno	114	78	68.42%
18	26-02-2019	421869	Diurno	111	77	69.37%
19	27-02-2019	352721	Diurno	103	73	70.87%
20	28-02-2019	352721	Diurno	107	74	69.16%



SANNA CLÍNICA

 Charlie Ramírez

 Jefe de Desarrollo de Sistemas

Nivel Porcentual de pacientes atendidos en triaje (Re-Test)

FICHA DE REGISTRO						
Investigadores	Oscar Garcia Del Aguila y Lincoln King Panizo Gonzales			Tipo de prueba	Re-Test	
Institución investigada	Clinica San Borja – Sanna					
Dirección	Av. Guardia Civil 337					
Motivo de Investigación	Nivel Porcentual de pacientes por atenderse en triaje					
Fecha Inicio	04 de marzo			Fecha Fin	29 de marzo	
Variable	Indicador	Medida		Formula		
Proceso de emergencia médica	Nivel Porcentual de pacientes atendidos en triaje	Porcentaje		$NPCA = \frac{CPC}{CTP} \cdot 100\%$		
Ítem	Fecha	Internista	Turno	Cantidad de pacientes admitidos (CTP)	Cantidad de pacientes atendidos en triaje(CPC)	Nivel Porcentual de pacientes atendidos en triaje (NPCA)
1	04-03-2019	365462	Diurno	111	77	69.37%
2	05-03-2019	352721	Diurno	110	76	69.09%
3	06-03-2019	365462	Diurno	109	77	70.64%
4	07-03-2019	352271	Diurno	108	74	68.52%
5	08-03-2019	396951	Diurno	103	71	68.93%
6	11-03-2019	352721	Diurno	112	77	68.75%
7	12-03-2019	365462	Diurno	111	75	67.57%
8	13-03-2019	421869	Diurno	112	77	68.75%
9	14-03-2019	421869	Diurno	110	76	69.09%
10	15-03-2019	352721	Diurno	107	75	70.09%
11	18-03-2019	365462	Diurno	107	76	71.03%
12	19-03-2019	352721	Diurno	103	70	67.96%
13	20-03-2019	365462	Diurno	105	70	66.67%
14	21-03-2019	483696	Diurno	103	70	67.96%
15	22-03-2019	352721	Diurno	105	72	68.57%
16	25-03-2019	365462	Diurno	112	77	68.75%
17	26-03-2019	365462	Diurno	115	78	67.83%
18	27-03-2019	352721	Diurno	112	77	68.75%
19	28-03-2019	476274	Diurno	105	73	69.52%
20	29-03-2019	352721	Diurno	109	75	68.81%


 Charlie Ramirez
 Jefe de Desarrollo de Sistemas

Nivel Porcentual de pacientes atendidos en triaje (Pre-Test)

FICHA DE REGISTRO						
Investigadores	Oscar Garcia Del Aguila y Lincoln King Panizo Gonzales			Tipo de prueba	Pre-Test	
Institución investigada	Clinica San Borja – Sanna					
Dirección	Av. Guardia Civil 337					
Motivo de Investigación	Nivel Porcentual de pacientes atendidos en triaje					
Fecha Inicio	03 de abril			Fecha Fin	30 de abril	
Variable	Indicador	Medida	Formula			
Proceso de emergencia médica	Nivel Porcentual de pacientes atendidos en traje	Porcentaje	$NPCA = \frac{CPC}{CTP} \times 100\%$			
Ítem	Fecha	Internista	Turno	Cantidad de pacientes admitidos (CTP)	Cantidad de pacientes atendidos en triaje (CPC)	Nivel Porcentual de pacientes atendidos en traje (NPCA)
1	03-04-2019	352721	Diurno	23	16	69,57%
2	04-04-2019	365462	Diurno	22	15	68,18%
3	05-04-2019	483696	Diurno	21	15	71,43%
4	08-04-2019	352721	Diurno	23	16	69,57%
5	09-04-2019	365462	Diurno	20	14	70,00%
6	10-04-2019	365462	Diurno	22	15	68,18%
7	11-04-2019	352721	Diurno	23	16	69,57%
8	12-04-2019	365462	Diurno	22	15	68,18%
9	15-04-2019	352271	Diurno	20	14	70,00%
10	16-04-2019	409609	Diurno	23	16	69,57%
11	17-04-2019	352721	Diurno	21	15	71,43%
12	18-04-2019	476274	Diurno	25	17	68,00%
13	19-04-2019	352721	Diurno	21	14	66,67%
14	22-04-2019	396951	Diurno	24	16	66,67%
15	23-04-2019	352721	Diurno	22	15	68,18%
16	24-04-2019	365462	Diurno	20	14	70,00%
17	25-04-2019	421869	Diurno	25	17	68,00%
18	26-04-2019	421869	Diurno	22	15	68,18%
19	29-04-2019	352721	Diurno	21	15	71,43%
20	30-04-2019	352721	Diurno	23	16	69,57%


SANNA Clínica
 Laboratorio
 Charlie Ramirez
 Jefe de Desarrollo de Sistemas

Nivel Porcentual de pacientes atendidos en triaje (Pos-Test)

FICHA DE REGISTRO						
Investigadores		Oscar Garcia Del Aguila y Lincoln King Panizo Gonzales		Tipo de prueba		Pos-Test
Institución investigada				Clinica San Borja – Sanna		
Dirección				Av. Guardia Civil 337		
Motivo de Investigación				Nivel Porcentual de pacientes atendidos en triaje		
Fecha Inicio		04 de noviembre		Fecha Fin		29 de noviembre
Variable		Indicador		Medida		Formula
Proceso de emergencia médica		Nivel Porcentual de pacientes atendidos en triaje		Porcentaje		<div>$NPCA = \frac{CPC}{CTP} \times 100\%$</div>
Ítem	Fecha	Internista	Turno	Cantidad de pacientes admitidos (CTP)	Cantidad de pacientes atendidos en triaje (CPC)	Nivel Porcentual de pacientes atendidos en triaje (NPCA)
1	04-11-2019	27	Diurno	17	16	94.12%
2	05-11-2019	26	Diurno	16	15	93.75%
3	06-11-2019	28	Diurno	20	19	95.00%
4	07-11-2019	29	Diurno	17	16	94.12%
5	08-11-2019	30	Diurno	15	14	93.33%
6	11-11-2019	30	Diurno	15	14	93.33%
7	12-11-2019	32	Diurno	16	15	93.75%
8	13-11-2019	31	Diurno	16	15	93.75%
9	14-11-2019	34	Diurno	15	14	93.33%
10	15-11-2019	27	Diurno	17	16	94.12%
11	18-11-2019	26	Diurno	19	18	94.74%
12	19-11-2019	30	Diurno	17	16	94.12%
13	20-11-2019	31	Diurno	13	12	92.31%
14	21-11-2019	32	Diurno	16	15	93.75%
15	22-11-2019	33	Diurno	16	15	93.75%
16	25-11-2019	35	Diurno	15	14	93.33%
17	26-11-2019	30	Diurno	15	14	93.33%
18	27-11-2019	31	Diurno	16	15	93.75%
19	28-11-2019	32	Diurno	19	18	94.74%
20	29-11-2019	33	Diurno	16	15	93.75%


 Charlie Ramirez
 Jefe de Desarrollo de Sistemas

Nivel Percentil de Cumplimiento (Test)

FICHA DE REGISTRO											
Investigadores		Oscar Garcia Del Aguila y Lincoln King Panizo Gonzales			Tipo de prueba		Test				
Institución investigada				Clinica San Borja – Sanna							
Dirección				Av. Guardia Civil 337							
Motivo de Investigación				Nivel Percentil de Cumplimiento							
Fecha Inicio		01 de febrero			Fecha Fin		28 de febrero				
Variable		Indicador			Medida		Formula				
Proceso de emergencia médica		Nivel percentil de cumplimiento			Porcentaje		$NPC = \frac{PAP}{TPAT} \cdot 100\%$				
Ítem	Fecha	Medico	Areas de Emergencia	Turno	Total, de pacientes admitidos (TPAT)	Pacientes atendidos por prioridad					Nivel Percentil de cumplimiento (NPC)
						P1	P2	P3	P4	P5	
1	01-02-2019	362764	Boxes	Diurno	110	1	6	17	38	0	56,36%
2	04-02-2019	531866	Boxes	Diurno	107	0	5	20	35	0	56,07%
3	05-02-2019	362764	Boxes	Diurno	108	0	3	19	39	0	56,48%
4	06-02-2019	484432	Boxes	Diurno	106	0	5	16	38	0	55,66%
5	07-02-2019	531866	Boxes	Diurno	104	1	6	16	35	0	55,77%
6	08-02-2019	490076	Boxes	Diurno	114	0	6	18	40	0	56,14%
7	11-02-2019	200849	Boxes	Diurno	112	0	4	24	35	0	56,25%
8	12-02-2019	3912023	Boxes	Diurno	114	0	5	23	35	0	55,26%
9	13-02-2019	304393	Boxes	Diurno	107	0	6	16	38	0	56,07%
10	14-02-2019	391203	Boxes	Diurno	107	1	5	18	36	0	56,07%
11	15-02-2019	362754	Boxes	Diurno	109	0	6	17	39	0	56,88%
12	18-02-2019	531866	Boxes	Diurno	103	0	5	14	39	0	56,31%
13	19-02-2019	490076	Boxes	Diurno	110	1	5	22	33	0	55,45%
14	20-02-2019	200849	Boxes	Diurno	103	0	6	15	37	0	56,31%
15	21-02-2019	3912023	Boxes	Diurno	105	0	7	14	36	0	54,29%
16	22-02-2019	304393	Boxes	Diurno	115	1	5	19	39	0	55,65%
17	25-02-2019	531866	Boxes	Diurno	114	0	6	20	36	0	54,39%
18	26-02-2019	490076	Boxes	Diurno	111	1	7	14	40	0	55,86%
19	27-02-2019	484432	Boxes	Diurno	103	0	6	19	33	0	56,31%
20	28-02-2019	490076	Boxes	Diurno	107	0	6	15	38	0	55,14%



SANNA CLÍNICA

SAN BORJA

 CHARLIE RAMÍREZ

 Jefe de Desarrollo de Sistemas

Nivel Percentil de Cumplimiento (Re – Test)

FICHA DE REGISTRO											
Investigadores		Oscar García Del Aguila y Lincoln King Panizo Gonzales			Tipo de prueba		Re-Test				
Institución investigada					Clínica San Borja – Sanna						
Dirección					Av. Guardia Civil 337						
Motivo de Investigación					Nivel Percentil de Cumplimiento						
Fecha Inicio		04 de marzo			Fecha Fin		29 de marzo				
Variable		Indicador			Medida		Formula				
Proceso de emergencia médica		Nivel percentil de cumplimiento			Porcentaje		$NPC = \frac{PAP}{TPAT} \cdot 100\%$				
Ítem	Fecha	Medico	Areas de Emergencia	Turno	Total, de pacientes admitidos (TPAT)	Pacientes atendidos por prioridad					Nivel Percentil de cumplimiento (NPC)
						P1	P2	P3	P4	P5	
1	04-03-2019	362764	Boxes	Diurno	111	0	7	18	37	0	55,86%
2	05-03-2019	531866	Boxes	Diurno	110	0	5	23	33	0	55,45%
3	06-03-2019	362764	Boxes	Diurno	109	1	3	20	38	0	56,88%
4	07-03-2019	484432	Boxes	Diurno	108	0	5	16	39	0	55,56%
5	08-03-2019	531866	Boxes	Diurno	103	1	6	16	35	0	56,31%
6	11-03-2019	490076	Boxes	Diurno	112	0	5	18	40	0	56,25%
7	12-03-2019	200849	Boxes	Diurno	111	0	5	24	33	0	55,86%
8	13-03-2019	3912023	Boxes	Diurno	112	0	5	23	34	0	55,36%
9	14-03-2019	304393	Boxes	Diurno	110	1	6	16	39	0	56,36%
10	15-03-2019	391203	Boxes	Diurno	107	0	5	19	36	0	56,07%
11	18-03-2019	362754	Boxes	Diurno	107	1	6	17	37	0	57,01%
12	19-03-2019	531866	Boxes	Diurno	103	0	5	14	39	0	56,31%
13	20-03-2019	490076	Boxes	Diurno	105	1	5	21	32	0	56,19%
14	21-03-2019	200849	Boxes	Diurno	103	0	6	15	37	0	56,31%
15	22-03-2019	3912023	Boxes	Diurno	105	0	6	14	37	0	54,29%
16	25-03-2019	304393	Boxes	Diurno	112	1	5	19	38	0	56,25%
17	26-03-2019	531866	Boxes	Diurno	115	0	6	20	37	0	54,78%
18	27-03-2019	490076	Boxes	Diurno	112	1	7	14	40	0	55,36%
19	28-03-2019	484432	Boxes	Diurno	105	0	5	19	34	0	55,24%
20	29-03-2019	490076	Boxes	Diurno	109	1	6	15	38	0	55,05%


SANNA CLÍNICA
 LABORATORIOS

 Charlie Ramirez
 Jefe de Desarrollo de Sistemas

Nivel Percentil de Cumplimiento (Pre – Test)

FICHA DE REGISTRO											
Investigadores		Oscar García Del Águila y Lincoln King Panizo Gonzales		Tipo de prueba		Pre-Test					
Institución investigada				Clínica San Borja – Sanna							
Dirección				Av. Guardia Civil 337							
Motivo de Investigación				Nivel Percentil de Cumplimiento							
Fecha Inicio		03 de abril		Fecha Fin		30de abril					
Variable		Indicador		Medida		Formula					
Proceso de emergencia médica		Nivel percentil de cumplimiento		Porcentaje		$NPC = \frac{PAP}{TPAT} * 100\%$					
Ítem	Fecha	Medico	Áreas de Emergencia	Turno	Total, de pacientes admitidos (TPAT)	Pacientes atendidos por prioridad					Nivel Percentil de cumplimiento o (NPC)
						P1	P2	P3	P4	P5	
1	03-04-2019	49914	Boxes	Diurno	23	1	2	5	7	0	65.22%
2	04-04-2019	531866	Boxes	Diurno	22	0	5	4	5	0	63.64%
3	05-04-2019	490076	Boxes	Diurno	21	0	5	4	5	0	66.67%
4	08-04-2019	484432	Boxes	Diurno	23	1	3	7	4	0	65.22%
5	09-04-2019	391203	Boxes	Diurno	20	0	3	3	7	0	65.00%
6	10-04-2019	362754	Boxes	Diurno	22	1	4	4	6	0	68.18%
7	11-04-2019	531866	Boxes	Diurno	23	0	3	7	5	0	65.22%
8	12-04-2019	490076	Boxes	Diurno	22	1	5	3	6	0	68.18%
9	15-04-2019	200849	Boxes	Diurno	20	0	3	5	5	0	65.00%
10	16-04-2019	3912023	Boxes	Diurno	23	2	3	5	5	0	65.22%
11	17-04-2019	304393	Boxes	Diurno	21	0	5	5	4	0	66.67%
12	18-04-2019	531866	Boxes	Diurno	25	2	4	5	6	0	68.00%
13	19-04-2019	1644	Boxes	Diurno	21	1	3	3	7	0	66.67%
14	22-04-2019	304393	Boxes	Diurno	24	1	2	7	6	0	66.67%
15	23-04-2019	484432	Boxes	Diurno	22	1	4	5	4	0	63.64%
16	24-04-2019	362764	Boxes	Diurno	20	1	2	4	6	0	65.00%
17	25-04-2019	531866	Boxes	Diurno	25	0	5	6	6	0	68.00%
18	26-04-2019	362764	Boxes	Diurno	22	1	5	4	5	0	68.18%
19	29-04-2019	484432	Boxes	Diurno	21	1	2	5	5	0	61.90%
20	30-04-2019	362764	Boxes	Diurno	23	2	2	5	6	0	65.22%


 Charile Ramirez
 Jefe de Desarrollo de Sistemas

Nivel Percentil de Cumplimiento (Pos – Test)

FICHA DE REGISTRO											
Investigadores		Oscar Garcia Del Aguila y Lincoln King Panizo Gonzales		Tipo de prueba		Pos-Test					
Institución investigada				Clinica San Borja – Sanna							
Dirección				Av. Guardia Civil 337							
Motivo de Investigación				Nivel Percentil de Cumplimiento							
Fecha Inicio		04 de noviembre		Fecha Fin		29 de noviembre					
Variable		Indicador		Medida		Formula					
Proceso de emergencia médica		Nivel percentil de cumplimiento		Porcentaje		$NPC = \frac{PAP}{TPAT} \cdot 100\%$					
Ítem	Fecha	Medico	Areas de Emergencia	Turno	Total, de pacientes admitidos (TPAT)	Pacientes atendidos por prioridad					Nivel Percentil de cumplimiento (NPC)
						P1	P2	P3	P4	P5	
1	04-11-2019	3	Boxes	Diurno	17	1	2	6	7	0	94.12%
2	05-11-2019	2	Boxes	Diurno	16	0	4	4	7	0	93.75%
3	06-11-2019	4	Boxes	Diurno	20	0	5	6	8	0	95.00%
4	07-11-2019	5	Boxes	Diurno	17	1	3	7	4	0	88.24%
5	08-11-2019	6	Boxes	Diurno	15	0	3	3	8	0	93.33%
6	11-11-2019	5	Boxes	Diurno	15	1	4	4	4	0	86.67%
7	12-11-2019	8	Boxes	Diurno	16	0	3	6	5	0	87.50%
8	13-11-2019	9	Boxes	Diurno	16	1	5	3	5	0	87.50%
9	14-11-2019	10	Boxes	Diurno	15	0	3	5	6	0	93.33%
10	15-11-2019	11	Boxes	Diurno	17	2	3	5	5	0	88.24%
11	18-11-2019	12	Boxes	Diurno	19	0	5	6	6	0	89.47%
12	19-11-2019	13	Boxes	Diurno	17	2	4	6	3	0	88.24%
13	20-11-2019	14	Boxes	Diurno	13	1	3	3	4	0	84.62%
14	21-11-2019	15	Boxes	Diurno	16	1	2	6	5	0	87.50%
15	22-11-2019	17	Boxes	Diurno	16	1	4	5	4	0	87.50%
16	25-11-2019	18	Boxes	Diurno	15	1	2	4	6	0	86.67%
17	26-11-2019	11	Boxes	Diurno	15	0	5	3	6	0	93.33%
18	27-11-2019	12	Boxes	Diurno	16	1	5	4	5	0	93.75%
19	28-11-2019	13	Boxes	Diurno	19	1	3	7	6	0	89.47%
20	29-11-2019	14	Boxes	Diurno	16	2	2	5	4	0	81.25%



 Charlie Ramírez

 Jefe de Desarrollo de Sistemas

Anexo 4: Base de Datos Experimental

ORDEN	NIVEL PORCENTUAL DE PACIENTES ATENDIDOS EN TRIAJE			NIVEL PERCENTIL DE CUMPLIMIENTO	
	Pre-test	Post-test		Pre-test	Post-test
1	0.70	0.89		0.65	0.83
2	0.68	0.88		0.64	0.81
3	0.71	0.94		0.67	0.88
4	0.70	0.89		0.65	0.83
5	0.70	0.93		0.65	0.87
6	0.68	0.88		0.68	0.88
7	0.70	0.89		0.65	0.83
8	0.68	0.88		0.68	0.88
9	0.70	0.93		0.65	0.87
10	0.70	0.89		0.65	0.83
11	0.71	0.94		0.67	0.88
12	0.68	0.85		0.64	0.85
13	0.67	0.88		0.67	0.88
14	0.67	0.89		0.67	0.89
15	0.68	0.88		0.68	0.88
16	0.70	0.93		0.65	0.87
17	0.68	0.90		0.68	0.85
18	0.68	0.88		0.64	0.82
19	0.71	0.94		0.67	0.88
20	0.70	0.89		0.65	0.83

Anexo 5: Resultados de la confiabilidad del instrumento

Indicador: Nivel Porcentual de Pacientes atendidos en triaje

	Xi	Yi	Xi * Yi	x^2 i	Y^2 i
Xi = Test Yi = ReTest	70,00	69,37	4855,90	4900,00	4812,20
	68,22	69,09	4713,32	4653,97	4773,43
	69,44	70,64	4905,24	4821,91	4990,01
	69,81	68,52	4783,38	4873,44	4694,99
	68,27	68,93	4705,85	4660,79	4751,34
	67,54	68,75	4643,38	4561,65	4726,56
	67,86	67,57	4585,30	4604,98	4565,70
	67,54	68,75	4643,38	4561,65	4726,56
	69,16	69,09	4778,26	4783,11	4773,43
	70,09	70,09	4912,61	4912,61	4912,61
	70,64	71,03	5017,56	4990,01	5045,26
	67,96	67,96	4618,56	4618,56	4618,56
	67,27	66,67	4484,89	4525,25	4444,89
	67,96	67,96	4618,56	4618,56	4618,56
	68,57	68,57	4701,84	4701,84	4701,84
	68,70	68,75	4723,13	4719,69	4726,56
	68,42	67,83	4640,93	4681,30	4600,91
	69,37	68,75	4769,19	4812,20	4726,56
	70,87	69,52	4926,88	5022,56	4833,03
	69,16	68,81	4758,90	4783,11	4734,82
Suma	1376,85	1376,65	94787,06	94807,18	94777,83

Media margin al de x	$\bar{x} = \sum_{i=1}^{20} x_i$	68,84
Media margin al de Y	$\bar{y} = \frac{\sum_{i=1}^{20} y_i}{20}$	68,83
Desviación típica marginal de X	$\sigma_x = \sqrt{\frac{\sum_{i=1}^{20} x_i^2}{10} - \bar{x}^2} =$	1.18
Desviación típica marginal de Y	$\sigma_y = \sqrt{\frac{\sum_{i=1}^{20} y_i^2}{10} - \bar{y}^2} =$	– 1.15
Covarianza	$\sigma_{xy} =$	1.0

	$\frac{\sum_{i=1}^{20} y_i}{10} - \bar{x}^2 \bar{y}^2$	
Coeficiente Correlación Pearson	$\gamma = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$	0.735

Correlaciones			
		TEST_NPCA	RETEST_NPCA
TEST_NPCA	Correlación de Pearson	1	,735**
	Sig. (bilateral)		,000
	N	20	20
RETEST_NPCA	Correlación de Pearson	,735**	1
	Sig. (bilateral)	,000	
	N	20	20

** . La correlación es significativa en el nivel 0,01 (bilateral).

Indicador: Nivel percentil de cumplimiento

Xi = Test Yi = ReTest	Xi	Yi	Xi * Yi	x^2 i	Y^2 i
	56,36	55,86	3148,27	3176,45	3120,34
	56,07	55,45	3109,08	3143,84	3074,70
	56,48	56,88	3212,58	3189,99	3235,33
	55,66	55,56	3092,47	3098,04	3086,91
	55,77	56,31	3140,41	3110,29	3170,82
	56,14	56,25	3157,88	3151,70	3164,06
	56,25	55,86	3142,13	3164,06	3120,34
	55,26	55,36	3059,19	3053,67	3064,73
	56,07	56,36	3160,11	3143,84	3176,45
	56,07	56,07	3143,84	3143,84	3143,84
	56,88	57,01	3242,73	3235,33	3250,14
	56,31	56,31	3170,82	3170,82	3170,82
	55,45	56,19	3115,74	3074,70	3157,32
	56,31	56,31	3170,82	3170,82	3170,82
	54,29	54,29	2947,40	2947,40	2947,40
	55,65	56,25	3130,31	3096,92	3164,06
	54,39	54,78	2979,48	2958,27	3000,85
	55,86	55,36	3092,41	3120,34	3064,73
	56,31	55,24	3110,56	3170,82	3051,46
	55,14	55,05	3035,46	3040,42	3030,50
Suma	1116,72	1116,75	62361,68	62361,58	62365,63

Media marginal de x	$= \sum_{i=1}^{20} x_i$	55.836
Media marginal de Y	$= \frac{\sum_{i=1}^{20} y_i}{20}$	55.8375
Desviación típica marginal de X	$\sigma_x = \sqrt{\frac{\sum_{i=1}^{20} x_i^2}{10} - \bar{x}^2} =$	0.6481
Desviación típica marginal de Y	$\sigma_y = \sqrt{\frac{\sum_{i=1}^{20} y_i^2}{10} - \bar{y}^2} =$	0.6746
Covarianza	$\sigma_{xy} = \frac{\sum_{i=1}^{20} x_i y_i}{10} - \bar{x}^2 \bar{y}^2$	0.34135
Coeficiente Correlación Pearson	$\gamma = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$	0.781

Correlaciones

		TEST_NP C	RETEST_NP C
TEST_NPC	Correlación de Pearson	1	,781**
	Sig. (bilateral)		,000
	N	20	20
RETEST_NP C	Correlación de Pearson	,781**	1
	Sig. (bilateral)	,000	
	N	20	20

**. La correlación es significativa en el nivel 0,01 (bilateral).

Anexo 6: Evaluación de las metodologías para el desarrollo

EVALUACION DE LA METODOLOGIA DE DESARROLLO DE SOFTWARE

TABLA EVALUACION DE EXPERTOS

Apellidos y nombres del experto: Díaz Reátegui, Mónica

Título y/o grado: Doctora

Fecha: 10/05/19

TITULO DE TESIS

SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO
DE EMERGENCIA MEDICA EN CLINICA SANNA SAN BORJA

EVALUACIÓN DE LA METODOLOGIA DE SOFTWARE

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar las metodologías involucradas, mediante una serie de criterios con puntuaciones especificadas al final de la tabla. Así mismo le exhortamos en la correcta determinación de la metodología para desarrollar el sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica en la empresa y si hubiese algunas sugerencias.

ITEM	CRITERIOS	Metodologías		
		OOHDM	UWE	XP
1	Resultados rápidos	3	2	1
2	Adaptabilidad	3	2	2
3	Asegura un software de alta calidad	3	1	1
4	Desarrollo orientado a objetos	3	2	2
5	Implementa las necesidades del sistema	3	2	2
6	Integra eficientemente todas las fases del ciclo de software	3	2	2
7	Desarrollo iterativo e incremental	3	2	2
8	Desarrollo a base de la comunicación con el usuario y retroalimentación.	3	2	2
	Total	24	15	14

La escala a evaluar es de 1: Malo, 2: Regular y 3: Bueno

Sugerencias:.....

.....

.....


Firma Experto

EVALUACION DE LA METODOLOGIA DE DESARROLLO DE SOFTWARE

TABLA EVALUACION DE EXPERTOS

Apellidos y nombres del experto: Gálvez Tapia Orleans

Título y/o grado: Magister en Ingeniería de Sistemas

Fecha: 10/05/2019

TITULO DE TESIS

SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO
DE EMERGENCIA MEDICA EN CLINICA SANNA SAN BORJA

EVALUACIÓN DE LA METODOLOGIA DE SOFTWARE

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar las metodologías involucradas, mediante una serie de criterios con puntuaciones especificadas al final de la tabla. Así mismo le exhortamos en la correcta determinación de la metodología para desarrollar el sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica en la empresa y si hubiese algunas sugerencias.

ITEM	CRITERIOS	Metodologías		
		OOHDM	UWE	XP
1	Resultados rápidos	3	3	2
2	Adaptabilidad	3	2	2
3	Asegura un software de alta calidad	3	2	2
4	Desarrollo orientado a objetos	3	3	2
5	Implementa las necesidades del sistema	3	2	2
6	Integra eficientemente todas las fases del ciclo de software	3	2	2
7	Desarrollo iterativo e incremental	3	2	2
8	Desarrollo a base de la comunicación con el usuario y retroalimentación.	3	2	2
	Total	24	18	16

La escala a evaluar es de 1: Malo, 2: Regular y 3: Bueno

Sugerencias:

.....

.....

Firma Experto

EVALUACION DE LA METODOLOGIA DE DESARROLLO DE SOFTWARE

TABLA EVALUACION DE EXPERTOS

Apellidos y nombres del experto: Cueva Villavicencio Soanila Isabel

Título y/o grado: Magister

Fecha: 09/05/2019

TITULO DE TESIS

SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO DE EMERGENCIA MEDICA EN CLINICA SANNA SAN BORJA

EVALUACIÓN DE LA METODOLOGIA DE SOFTWARE

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar las metodologías involucradas, mediante una serie de criterios con puntuaciones especificadas al final de la tabla. Así mismo le exhortamos en la correcta determinación de la metodología para desarrollar el sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica en la empresa y si hubiese algunas sugerencias.

ITEM	CRITERIOS	Metodologías		
		OOHDM	UWE	XP
1	Resultados rápidos	2	2	3
2	Adaptabilidad	3	3	3
3	Asegura un software de alta calidad	2	3	1
4	Desarrollo orientado a objetos	3	2	2
5	Implementa las necesidades del sistema	3	3	2
6	Integra eficientemente todas las fases del ciclo de software	3	2	2
7	Desarrollo iterativo e incremental	3	3	3
8	Desarrollo a base de la comunicación con el usuario y retroalimentación.	2	2	3
	Total	21	20	19

La escala a evaluar es de 1: Malo, 2: Regular y 3: Bueno

Sugerencias:.....

.....

.....


Firma Experto

TABLA DE EVALUACION DE EXPERTOS

Apellidos y Nombres del Experto: Cueva Villavicencio Juanita Isabel
 Titulo y/o grado: Magister
 Universidad donde labora: Universidad Cesar Vallejo
 Fecha:

TITULO DE PROYECTO

SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO DE EMERGENCIA MEDICA EN CLINICA SANNA SAN BORJA

Tabla de Evaluación de Expertos para el indicador: Nivel porcentual de pacientes atendidos

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de criterios marcando un valor en las columnas. Asimismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias.

ITEMS	CRITERIOS	VALORACION				
		Deficiente 0-20%	Regular 21-50%	Bueno 51-70%	Muy Bueno 71-80%	Excelente 81-100%
CLARIDAD	Esta formado por el lenguaje adecuado					85%
OBJETIVIDAD	Esta expresado en conducta observable					85%
ORGANIZACIÓN	Es adecuado al avance de la ciencia y tecnología					85%
SUFICIENCIA	Comprende los aspectos de cantidad y calidad					85%
INTENCIONALIDAD	Adecuado para valorar los aspectos del sistema metodológico y científico					85%
CONSISTENCIA	Esta basado en aspectos técnicos, científicos acordes a la tecnología adecuada					85%
COHERENCIA	Entre los índices, indicadores y dimensiones					85%
METODOLOGIA	Responde el propósito del trabajo bajo los objetivos a lograr					85%
PERTINENCIA	El instrumento es adecuado al tipo de investigación					85%

PROMEDIO DE VALORACION:

OPCION DE APLICABILIDAD:

- () El instrumento puede ser aplicado, tal como está elaborado.
 () El instrumento debe ser mejorado, antes de ser aplicado.


 Firma Experto

TABLA DE EVALUACION DE EXPERTOS

Apellidos y Nombres del Experto: Ornelio Rojas, Robert
 Título y/o grado: Magister
 Universidad donde labora: DIP-UCV
 Fecha: 14/06/2019

TITULO DE PROYECTO

SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO DE EMERGENCIA MEDICA EN CLINICA SANNA SAN BORJA

Tabla de Evaluación de Expertos para el indicador: Nivel porcentual de pacientes atendidos

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de criterios marcando un valor en las columnas. Asimismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias.

ITEMS	CRITERIOS	VALORACION				
		Deficiente 0-20%	Regular 21-50%	Bueno 51-70%	Muy Bueno 71-80%	Excelente 81-100%
CLARIDAD	Esta formado por el lenguaje adecuado				80%	
OBJETIVIDAD	Esta expresado en conducta observable				80%	
ORGANIZACIÓN	Es adecuado al avance de la ciencia y tecnología				80%	
SUFICIENCIA	Comprende los aspectos de cantidad y calidad				80%	
INTENCIONALIDAD	Adecuado para valorar los aspectos del sistema metodológico y científico				80%	
CONSISTENCIA	Esta basado en aspectos técnicos, científicos acordes a la tecnología adecuada				80%	
COHERENCIA	Entre los índices, indicadores y dimensiones				80%	
METODOLOGIA	Responde el propósito del trabajo bajo los objetivos a lograr				80%	
PERTINENCIA	El instrumento es adecuado al tipo de investigación				80%	

PROMEDIO DE VALORACION: 80%

OPCION DE APLICABILIDAD:

☒ El instrumento puede ser aplicado, tal como está elaborado.

☐ El instrumento debe ser mejorado, antes de ser aplicado.


 Firma Experto

TABLA DE EVALUACION DE EXPERTOS

Apellidos y Nombres del Experto: Gordillo Huamanchumo Luis A.
 Titulo y/o grado: Magister
 Universidad donde labora: UCV - UTP
 Fecha: 12/06/19

TITULO DE PROYECTO

SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO DE EMERGENCIA MEDICA EN CLINICA SANNA SAN BORJA

Tabla de Evaluación de Expertos para el indicador: Nivel porcentual de pacientes atendidos

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de criterios marcando un valor en las columnas. Asimismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias.

ITEMS	CRITERIOS	VALORACION				
		Deficiente 0-20%	Regular 21-50%	Bueno 51-70%	Muy Bueno 71-80%	Excelente 81-100%
CLARIDAD	Esta formado por el lenguaje adecuado				78%	
OBJETIVIDAD	Esta expresado en conducta observable				78%	
ORGANIZACIÓN	Es adecuado al avance de la ciencia y tecnología				78%	
SUFICIENCIA	Comprende los aspectos de cantidad y calidad				78%	
INTENCIONALIDAD	Adecuado para valorar los aspectos del sistema metodológico y científico				78%	
CONSISTENCIA	Esta basado en aspectos técnicos, científicos acordes a la tecnología adecuada				78%	
COHERENCIA	Entre los índices, indicadores y dimensiones				78%	
METODOLOGIA	Responde el propósito del trabajo bajo los objetivos a lograr				78%	
PERTINENCIA	El instrumento es adecuado al tipo de investigación				78%	

PROMEDIO DE VALORACION: 78%

OPCION DE APLICABILIDAD:

☒ El instrumento puede ser aplicado, tal como está elaborado.

☐ El instrumento debe ser mejorado, antes de ser aplicado.



Firma Experto

TABLA DE EVALUACION DE EXPERTOS

Apellidos y Nombres del Experto: Ormeau Rojas, Robert Eduardo

Título y/o grado: Magister

Universidad donde labora:

Fecha: 14/06/2012

TITULO DE PROYECTO

SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO DE EMERGENCIA MEDICA EN CLINICA SANNA SAN BORJA

Tabla de Evaluación de Expertos para el indicador: Nivel percentil de cumplimiento

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de criterios marcando un valor en las columnas. Asimismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias.

ITEMS	CRITERIOS	VALORACION				
		Deficiente 0-20%	Regular 21-50%	Bueno 51-70%	Muy Bueno 71-80%	Excelente 81-100%
CLARIDAD	Esta formado por el lenguaje adecuado				80%	
OBJETIVIDAD	Esta expresado en conducta observable				80%	
ORGANIZACIÓN	Es adecuado al avance de la ciencia y tecnología				80%	
SUFICIENCIA	Comprende los aspectos de cantidad y calidad				80%	
INTENCIONALIDAD	Adecuado para valorar los aspectos del sistema metodológico y científico				80%	
CONSISTENCIA	Esta basado en aspectos técnicos, científicos acordes a la tecnología adecuada				80%	
COHERENCIA	Entre los índices, indicadores y dimensiones				80%	
METODOLOGIA	Responde el propósito del trabajo bajo los objetivos a lograr				80%	
PERTINENCIA	El instrumento es adecuado al tipo de investigación				80%	

PROMEDIO DE VALORACION: 80%

OPCION DE APLICABILIDAD:

☒ El instrumento puede ser aplicado, tal como está elaborado.

☐ El instrumento debe ser mejorado, antes de ser aplicado.


Firma Experto

TABLA DE EVALUACION DE EXPERTOS

Apellidos y Nombres del Experto: Gordillo Huamanchumo Luis A.
 Titulo y/o grado: Magister
 Universidad donde labora: UCV - UTP
 Fecha: 12-06-19.

TITULO DE PROYECTO

SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO DE EMERGENCIA
 MEDICA EN CLINICA SANNA SAN BORJA

Tabla de Evaluación de Expertos para el indicador: Nivel percentil de cumplimiento

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de criterios marcando un valor en las columnas. Asimismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias.

ITEMS	CRITERIOS	VALORACION				
		Deficiente 0-20%	Regular 21-50%	Bueno 51-70%	Muy Bueno 71-80%	Excelente 81-100%
CLARIDAD	Esta formado por el lenguaje adecuado				80%	
OBJETIVIDAD	Esta expresado en conducta observable				80%	
ORGANIZACIÓN	Es adecuado al avance de la ciencia y tecnología				80%	
SUFICIENCIA	Comprende los aspectos de cantidad y calidad				80%	
INTENCIONALIDAD	Adecuado para valorar los aspectos del sistema metodológico y científico				80%	
CONSISTENCIA	Esta basado en aspectos técnicos, científicos acordes a la tecnología adecuada				80%	
COHERENCIA	Entre los índices, indicadores y dimensiones				80%	
METODOLOGIA	Responde el propósito del trabajo bajo los objetivos a lograr				80%	
PERTINENCIA	El instrumento es adecuado al tipo de investigación				80%	

PROMEDIO DE VALORACION: 80%

OPCION DE APLICABILIDAD:

☒ El instrumento puede ser aplicado, tal como está elaborado.

☐ El instrumento debe ser mejorado, antes de ser aplicado.


Firma Experto

TABLA DE EVALUACION DE EXPERTOS

Apellidos y Nombres del Experto: Cuervo Villanueva Benito Gabriel
Titulo y/o grado: Magister
Universidad donde labora: Universidad Cesar Vallejo
Fecha:

TITULO DE PROYECTO

SISTEMA WEB UTILIZANDO EL PROTOCOLO DE MANCHESTER PARA EL PROCESO DE EMERGENCIA MEDICA EN CLINICA SANNA SAN BORJA

Tabla de Evaluación de Expertos para el indicador: Nivel percentil de cumplimiento

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de criterios marcando un valor en las columnas. Asimismo, le exhortamos en la corrección de los ítems indicando sus observaciones y/o sugerencias.

ITEMS	CRITERIOS	VALORACION				
		Deficiente 0-20%	Regular 21-50%	Bueno 51-70%	Muy Bueno 71-80%	Excelente 81-100%
CLARIDAD	Esta formado por el lenguaje adecuado					85%
OBJETIVIDAD	Esta expresado en conducta observable					85%
ORGANIZACIÓN	Es adecuado al avance de la ciencia y tecnología					85%
SUFICIENCIA	Comprende los aspectos de cantidad y calidad					85%
INTENCIONALIDAD	Adecuado para valorar los aspectos del sistema metodológico y científico					85%
CONSISTENCIA	Esta basado en aspectos técnicos, científicos acordes a la tecnología adecuada					85%
COHERENCIA	Entre los índices, indicadores y dimensiones					85%
METODOLOGIA	Responde el propósito del trabajo bajo los objetivos a lograr					85%
PERTINENCIA	El instrumento es adecuado al tipo de investigación					85%

PROMEDIO DE VALORACION:

OPCION DE APLICABILIDAD:

- ☐ El instrumento puede ser aplicado, tal como está elaborado.
☐ El instrumento debe ser mejorado, antes de ser aplicado.


Firma Experto

Anexo 7: Entrevista

ENCUESTA PARA ANALIZAR Y DETERMINAR LA PROBLEMÁTICA ACTUAL DE LA CLINICA SAN BORJA – SANNA EN EL AREA DE EMERGENCIA

Nombre del entrevistado: Charlie Ramírez Agurto
Cargo del entrevistador: Jefe de Desarrollo de Sistemas
Departamento/Área: Desarrollo de Sistemas

Instrucciones

Sírvase de contestar las siguientes preguntas, por lo que se espera un alto grado de sinceridad, ya que su acelerada respuesta contribuirá con el desarrollo de la organización y de esta manera se podrá mejorar los procesos y el ambiente laboral.

1) ¿Realice un resumen general del proceso de emergencia (desde la admisión hasta el alta)?

El proceso médico de emergencia inicia cuando el paciente ingresa debido a algún motivo por el cual debe ser atendido, lo primero a realizar es la admisión de la atención para lo cual el paciente o acompañante debe entregar el documento de identidad u otro documento de identidad, luego de haber registrado al paciente este es derivado al área de triaje, si el paciente ingresa por un motivo muy grave se deriva directo al tópico de trauma shock y el registro de los datos de la atención se ingresan después de haber atendido este caso. Al tener ya la orden de atención como ya se mencionó el paciente debe dirigirse al área de triaje en el cual se evaluará sus signos vitales, se ingresa el motivo de ingreso, antecedentes, alergias, box correspondiente para la atención de emergencia y se asigna la prioridad del 1 al 4, además se le coloca al paciente una cinta de color de acuerdo a la prioridad asignada, si tuviese alergia se le coloca una cinta roja adicional. Después de pasar por triaje la enfermera guía al paciente al box correspondiente a su motivo de ingreso ya sea: pediátrico, adultos, traumatología, otros. En el box asignado debe atender el médico de turno o especialista. El médico de turno o especialista se asigna a la atención y solo él puede realizar la atención a no ser que realice una interconsulta. Al ingresar a la atención el médico registra en los formatos que posee el sistema los cuales son: Anamnesis, antecedentes, alergias, diagnóstico, receta, exámenes, descanso médico e interconsulta. En el caso de las interconsultas existen de tres tipos como la interconsulta por manejo en conjunto, opinión y derivación. El médico encargado es el que realiza el alta de emergencia para poder dar fin a la atención, pero no podrá


Sanna Clínica
San Borja
Charlie Ramírez
Jefe de Desarrollo de Sistemas

realizar el alta si es que la enfermera no termina con su atención. En el caso de la enfermera, también realiza su propio registro en unas fichas nombradas como notas de enfermería en las cuales indica la evolución del paciente como si fuese un historial, además añade algún medicamento en la ficha si fuese el caso. Al realizar el registro de la atención por parte de la enfermera, ya el médico puede realizar el alta del paciente y brinda una receta de alta o un descanso médico si es que fuese el caso.

2) ¿Ha encontrado ciertos problemas desde la admisión a triaje?

Algunos casos el paciente no lleva consigo el documento que lo pueda identificar para estos casos se toma en consideración otro tipo de documento que lo identifique o se anota los datos necesarios es una ficha, pero este es un problema ya que genera demora para poder registrarlo por sistema ya que el DNI es una información necesaria para poder registrarlo en el sistema actual que maneja la clínica, este es un sistema de escritorio el cual no está programado para poder realizarle cambios.

3) ¿Ha encontrado ciertos problemas en el área de triaje?

Actualmente los datos de la atención del triaje se ingresan de forma manual en una ficha y se adjunta en el historial de la atención. Un problema fundamental en este módulo es que las fichas de registro de triaje pueden ser extraviadas o en algunos casos la información no es la correcta y esto surge problemas ya en la atención de la emergencia, debido a que genera demora en el tiempo para poder atender al paciente.

4) ¿Ha encontrado ciertos problemas desde el triaje a emergencia?

Actualmente son los internistas los que ingresan al sistema de escritorio la asignación de la prioridad y el box que le corresponde al paciente para que luego el médico de turno o especialista pueda visualizar en la pantalla a los pacientes que tienen en cola, otro problema en esta fase es que los internistas pueden ingresar datos erróneos y no se lleva un control de los boxes ocupados en el momento por parte del sistema actual. además, que al ingresar mal un dato fundamental como es la prioridad puede llevar al error de lograr una atención rápida y aquí es donde se genera la mayor molestia por parte del paciente, ya que en algunos casos deben esperar a ser atendidos más del tiempo que les corresponde, además en la pantalla que visualiza el médico observa que sale ordenado de acuerdo al tiempo de ingreso y no a la prioridad de atención,

este cambio no se puede realizar debido a que el sistema actual no lo permite. Los médicos por su parte no se guían mucho por el orden mostrado en la pantalla; sino de la prioridad, pero sería de mucha ayuda si es que se pudiese ordenar de acuerdo a la prioridad.

5) ¿Cuáles son los tipos de interconsulta y como se manejan?

La interconsulta por “manejo en conjunto” en este tipo de interconsulta el médico realiza la atención desde su plantilla y el otro médico asignado realiza su atención en una nueva plantilla. En la interconsulta por “opinión” el médico opta por tener una opinión del caso tratado de parte de otro médico especialista, este tendrá solo un acceso a la atención. La interconsulta por “derivación” el médico deriva el caso con otro médico especialista, este tiene el acceso a toda la atención y se encargará de dar el alta médica.

6) ¿Cuáles son las actividades de la enfermera?

Realiza su propio registro en unas fichas nombradas como notas de enfermería en las cuales indica la evolución del paciente como si fuese un historial, además añade algún medicamento en la ficha si fuese el caso

7) ¿Existe casos de abandono y como se manejan estos?

Si, solo se indica de forma manual en las fichas de registro que el paciente ha abandonado la atención o en el sistema los médicos registran en el campo de relato del formulario de anamnesis que el paciente ha abandonado la atención de forma voluntaria.

8) ¿Cuál cree que es el problema fundamental de todo el proceso de emergencia?

En general, se podría decir que los cuellos de botella que existe desde la admisión hasta la atención médica donde el intermediario es el área de triaje.

Anexo 8: Carta de aprobación de la empresa

Carta de Aceptación

“Sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica en la Clínica San Borja – Sanna”

Mediante este documento se certifica:

Que el Sr. Oscar Garcia Del Aguila, identificado con DNI: 47772043, estudiante de la Escuela de Ingeniería de Sistemas de la Universidad César Vallejo, ha sido aceptado por nuestra institución para realizar el proyecto de investigación dentro de las instalaciones del área de desarrollo, dando como conformidad que la Clínica San Borja – Sanna brindara toda la información necesaria para la elaboración de la presente investigación de un “Sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica en la Clínica San Borja – Sanna”.

Como condiciones contractuales, el estudiante se obliga no divulgar ni usar para fines personales la información, con objeto de la relación de trabajo, que le fue suministrada; no proporcionar a terceras personas, verbalmente o por escrito, directa o indirectamente, información alguna de las actividades y/o procesos de cualquier clase que fuesen observadas en la institución por políticas de seguridad. El estudiante que asume toda la información será de uso exclusivamente para el desarrollo de la presente investigación.

Se expresa el agradecimiento y se expide el documento de acuerdo lo solicitado del interesado para los fines que él lo requiera.

San Borja, 03 de abril de 2019


Charlie Ramirez
Jefe de Desarrollo de Sistemas

Anexo 9 Metodología OOHDM

PRESENTACIÓN

La Clínica Sanna San Borja, ubicada en Av. Guardia Civil 337 es una entidad con una infraestructura amplia y a la vanguardia para brindar el mejor servicio para la salud del paciente y buscar siempre su bienestar. La visión fundamental de la clínica es ser una de las entidades de salud más conocidas del país obteniendo los estándares de calidad y seguridad médica. El área de emergencia cuenta con un servicio implementado con todos los equipos necesarios para una atención oportuna, cuenta con una unidad de Trauma - shock, tópico para cirugía menor – Poli traumatizados, así también atención médica en Pediatría, medicina interna, cirugía y Cardiología, etc.

Para el desarrollo de software de la presente investigación se utilizó la metodología OOHDm, asimismo la codificación se realizará a través se utilizó el lenguaje de programación Java con el Spring framework 3.9.2 y el gestor de base de datos SQL Server 2012 R2.

La metodología escogida para el desarrollo del Sistema web fue OOHDm, esta metodología está orientada al desarrollo web con el principal propósito de brindar una interfaz amigable a sus futuros usuarios para cumplir con los reglamentos necesarios que la entidad gubernamental nos brinda en los tiempos pactados. La metodología consta de 5 fases: Obtención de requerimientos, Modelo Conceptual, Diseño Navegacional, Diseño de Interfaz e Implementación; permitiendo desarrollar el Sistema web utilizando el protocolo de Manchester para el proceso de emergencia médica en la clínica Sanna San Borja.

Fase 1: Obtención de Requerimientos

En esta fase de la metodología OOHDM se evidencia los casos de uso, los cuales han sido diagramados por perfiles con el objetivo de obtener los requerimientos y eventos del sistema, para obtener esto se definieron los requerimientos funcionales y no funcionales, la relación de los casos de uso y requerimientos funcionales, diagramas de casos de uso, se identificó los roles y tareas, especificación de escenarios, casos de uso e interacción con el usuario.

Requerimientos Funcionales

En la tabla N° 15, se muestra la lista de requerimientos funcionales que han sido identificados para el desarrollo del proyecto mediante un levantamiento de información con los usuarios involucrados al proceso de emergencia médica.

Tabla 15 Requerimiento Funcionales

Código	Requerimiento Funcional	Descripción	Prioridad
RF01	Ingresa al sistema	El sistema debe permitir el inicio de sesión ingresando un usuario, contraseña y tipo de usuario para su autenticación.	ALTA
RF02	Gestionar usuario	El sistema debe permitir al usuario admisionista registrar y actualizar los distintos tipos de usuarios que se interrelacionan con el sistema web.	ALTA
RF03	Gestionar paciente	El sistema debe permitir al usuario admisionista registrar y actualizar pacientes.	ALTA
RF04	Crear orden atención	El sistema debe permitir al usuario admisionista la creación de las ordenes de atención del paciente.	ALTA
RF05	Listar orden atención	El sistema debe listar las ordenes de atención en cola en la bandeja de atención correspondiente a cada usuario, para el usuario internista se listará con los estados pendiente y en atención, pero para los usuarios médico y enfermera se listará con los estados pendiente, en atención y atendido,	ALTA

		además debe estar ordenados por las prioridades de atención en triaje.	
RF06	Listar motivo atención	El sistema debe permitir al usuario internista listar los motivos de atención de acuerdo al protocolo de Manchester.	ALTA
RF07	Registrar triaje	El sistema debe permitir al usuario internista el registro de triaje.	ALTA
RF08	Actualizar abandono	El sistema debe permitir a los usuarios internista, médico y enfermera actualizar el estado de atención de triaje, emergencia o de enfermería respectivamente de la orden atención a estado de abandono si es que un paciente no se presenta a la atención.	ALTA
RF09	Generar reporte nivel porcentual de pacientes atendidos en triaje	El sistema debe permitir al usuario internista generar un reporte y exportarlo a un formato Excel.	ALTA
RF10	Generar reporte triaje	El sistema debe permitir al usuario internista generar un reporte y exportarlos a un formato Excel.	ALTA
RF11	Gestionar Enfermedad Actual	El sistema debe permitir al usuario médico poder registrar y actualizar el formato de enfermedad actual para el registro de emergencia.	ALTA
RF12	Gestionar Funciones Vitales	El sistema debe permitir al usuario médico poder registrar y actualizar el formato de funciones vitales para el registro de emergencia.	ALTA
RF13	Gestionar diagnostico	El sistema debe permitir al usuario médico poder registrar, eliminar y actualizar el formato de funciones vitales para el registro de emergencia.	ALTA
RF14	Gestionar exámenes de apoyo al diagnóstico	El sistema debe permitir al usuario médico poder registrar, eliminar y actualizar el	ALTA

		formato de exámenes de apoyo al diagnóstico para el registro de emergencia.	
RF15	Gestionar receta	El sistema debe permitir al usuario médico poder registrar, eliminar y actualizar el formato de receta para el registro de emergencia.	ALTA
RF16	Gestionar descanso medico	El sistema debe permitir al usuario médico poder registrar y actualizar el formato de descanso médico para el registro de emergencia.	ALTA
RF17	Actualizar alta medica	El sistema debe permitir al usuario médico poder actualizar el estado de emergencia de la orden atención a atendido siempre y cuando exista un diagnóstico de atención y que el usuario enfermera haya actualizado su acto de enfermería a estado atendido.	ALTA
RF18	Generar reporte nivel percentil de cumplimiento	El sistema debe permitir al usuario médico generar un reporte y exportarlo a un formato Excel.	ALTA
RF19	Generar reporte emergencia	El sistema debe permitir al usuario médico generar un reporte y exportarlo a un formato Excel.	ALTA
RF20	Generar reporte de atención emergencia	El sistema debe permitir al usuario médico imprimir un formato pdf de la atención de emergencia.	ALTA
RF21	gestionar notas enfermería	El sistema debe permitir al usuario enfermera poder registrar y editar el formato notas de enfermería.	ALTA
RF22	Actualizar acto enfermería	El sistema debe permitir al usuario enfermera poder actualizar el estado de enfermería de la orden atención a atendido siempre y cuando exista registro del formato notas de enfermería.	ALTA

Fuente: Elaboración propia

Como se muestra en la tabla N°14 se han obtenido 22 requerimientos funcionales para la implementación del sistema web utilizando el protocolo de Manchester para su correcto funcionamiento.

Requerimientos No Funcionales

En la tabla N° 16, se muestra el listado de los requerimientos no funcionales que han sido identificados para el desarrollo del proyecto mediante un levantamiento de información con los usuarios involucrados al proceso de emergencia médica.

Tabla 16 Requerimientos No Funcionales

Código	Requerimiento No Funcional	Prioridad
RNF01	El sistema debe tener en cuenta cada perfil de usuario y sus accesos.	ALTA
RNF02	El sistema debe ser escalable y permitir a futuro la implementación de nuevas funcionalidades.	ALTA
RNF03	El sistema debe arrojar mensajes de error.	ALTA
RNF04	El sistema debe arrojar mensajes de satisfacción.	ALTA
RNF05	El sistema debe ser desarrollado en un entorno web, con lenguaje java y con un gestor de base de datos SQL Server	ALTA
RNF06	El sistema debe contener los colores institucionales y el logo que representan a la empresa	ALTA






Fuente: Elaboración propia










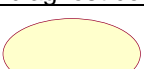
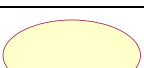
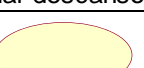
Como se muestra en la tabla N°15 se han obtenido 6 requerimientos no funcionales para la implementación del sistema web utilizando el protocolo de Manchester para su correcto funcionamiento.






Relación entre los casos de uso y los requerimientos funcionales

En la tabla N° 17, se muestra la relación entre los casos de uso y los requerimientos funcionales, además se identifica el actor que se relaciona con cada requerimiento y caso de uso.

Tabla 17 Relación entre los casos de uso y los requerimientos funcionales

Código	Nombre de los Casos de Uso	Actor	Requerimiento Funcional
CUS01	 ingresar al sistema	Admisionista, Internista, Médico y Enfermera	RF01
CUS02	 Gestionar usuario	Admisionista	RF02
CUS03	 Gestionar paciente	Admisionista	RF03
CUS04	 crear orden atencion	Admisionista	RF04
CUS05	 Listar orden atencion	Admisionista, Internista, Médico y Enfermera	RF05

CUS06	 listar motivo atención	Internista	RF06
CUS07	 registrar triaje	Internista	RF07
CUS08	 actualizar abandono	Internista, Médico y Enfermera	RF08
CUS09	 generar reporte nivel porcentual de pacientes atendidos en triaje	Internista	RF09
CUS10	 generar reporte triaje	Internista	RF10
CUS11	 gestionar enfermedad actual	Médico	RF11
CUS12	 gestionar funciones vitales	Médico	RF12
CUS13	 gestionar diagnostico	Médico	RF13
CUS14	 gestionar exámenes de apoyo al diagnostico	Médico	RF14
CUS15	 gestionar receta	Médico	RF15
CUS16	 gestionar descanso medico	Médico	RF16
CUS17	 actualizar alta medica	Médico	RF17

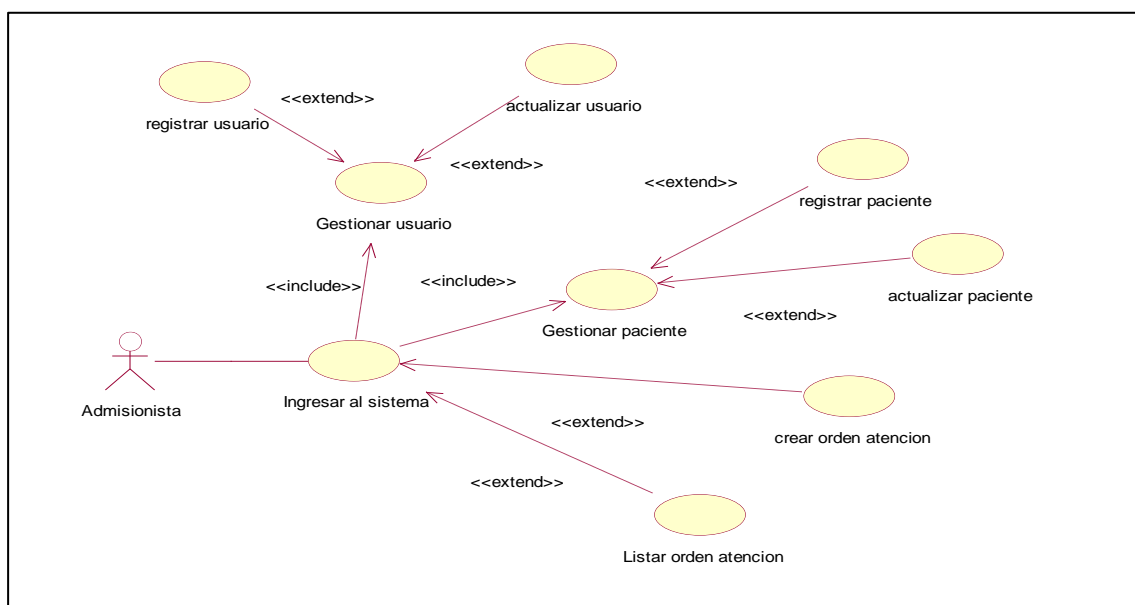
CUS18	 generar reporte nivel percentil de cumplimiento	Médico	RF18
CUS19	 generar reporte emergencia	Médico	RF19
CUS20	 generar reporte de atención emergencia	Médico	RF20
CUS21	 gestionar notas enfermería	Enfermera	RF21
CUS22	 actualizar acto enfermería	Enfermera	RF22

Fuente: Elaboración propia

Diagrama de Caso de Uso del Sistema

Perfil 1: Admisionistas. En la Figura 22 podemos observar la iteración del Admisionista con el sistema web para ingresar al sistema y tener la posibilidad de gestionar usuarios y pacientes, además podrá realizar la creación de orden de atención y poder listar en una bandeja las ordenes creadas.

Figura 22 Caso de uso admisionista

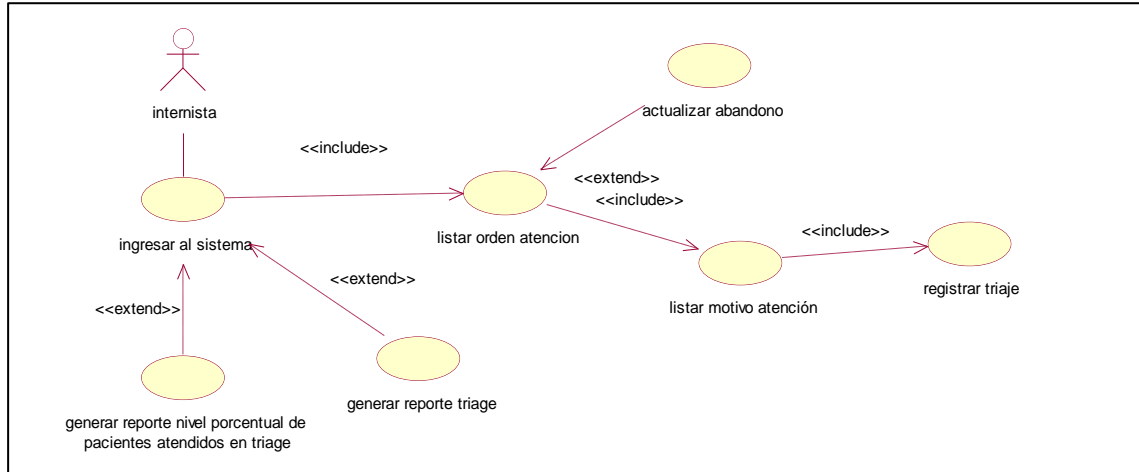


Fuente: Elaboración propia

Perfil 2: Internista.

En la Figura 23 podemos observar la iteración del internista con el sistema web para ingresar al sistema y tener el control del módulo de triaje donde listará, abandonará y podrá registrar el triaje, además podrá generar reportes.

Figura 23 Caso de uso internista

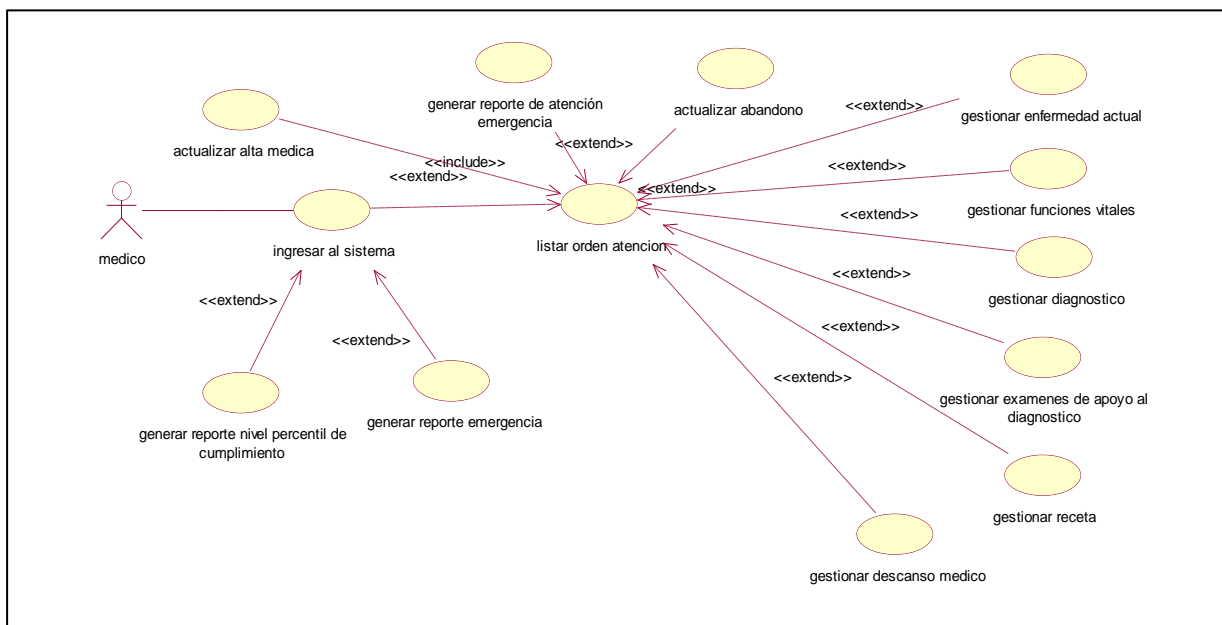


Fuente: Elaboración propia

Perfil 3: Medico.

En la Figura 24 podemos observar la iteración del médico con el sistema web para ingresar al sistema y podrá realizar la gestión de los distintos formatos que conforma la atención de emergencia, abandonar, listar orden atención, dar alta médica y generar reporte.

Figura 24 Caso de uso medico

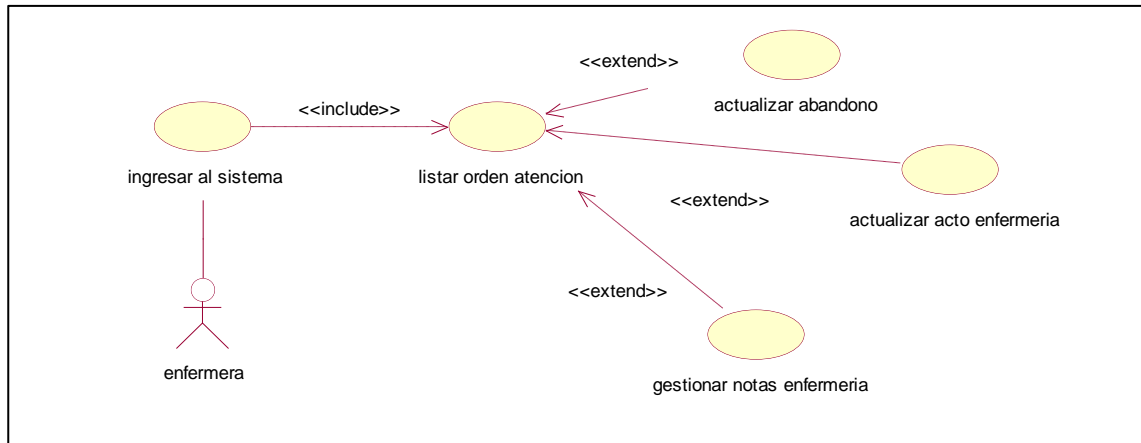


Fuente: Elaboración propia

Perfil 4: Enfermera.

En la Figura 25 podemos observar la iteración de la enfermera con el sistema web para ingresar al sistema y podrá realizar la gestión de las notas de enfermería, abandonar, lista orden atención y realizar su acto enfermería.

Figura 25 Caso de uso enfermera



Fuente: Elaboración propia

Identificaciones de roles y tareas

Se describe las necesidades del usuario en un documento con la finalidad de tener constancia de ellas a lo largo de todo el proceso de desarrollo. Estas necesidades son captadas a partir de las tareas que los diferentes usuarios deben poder realizar.

Los perfiles del usuario son los siguientes:

Perfil 1: Usuario Admisionista:

Es el usuario encargado del mantenimiento de pacientes y usuarios (médicos, internistas, enfermeras), además realizan la creación de las órdenes de atención, estos usuarios también tendrán el listado de los pacientes que están en cola en triaje para poder visualizar la demanda de atenciones.

- ✓ Ingresar al Sistema.
- ✓ Gestionar Usuario.
- ✓ Gestionar Paciente.
- ✓ Crear Orden Atención.
- ✓ Listar Orden Atención.

Perfil 2: Internistas:

Es el usuario encargado de las atenciones en el área de triaje, estos usuarios podrán visualizar las atenciones de triaje en estado pendiente y en atención en su bandeja de atenciones, también son los encargados del registro de triaje en donde le listara los motivos de atención y asignara la prioridad de atención para luego derivarlo a emergencia. Existen casos en los que los pacientes abandonan la atención así que el internista tendrá la funcionalidad de registrar el abandono de pacientes para liberar la atención, además podrá exportar reportes en formato Excel.

- ✓ Ingresar al Sistema.
- ✓ Listar Orden Atención.
- ✓ Listar Motivo de Atención
- ✓ Registrar triaje.
- ✓ Actualizar Abandono.
- ✓ Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage.
- ✓ Generar Reporte Triage.

Perfil 3: Medico:

Es el usuario encargado de registrar la atención de emergencia, este usuario puede visualizar sus atenciones en la bandeja de emergencia en todos los estados (pendiente, en atención y atendido), además tiene como tarea brindar el alta médica de la atención, esta función solo se habilita si la enfermera ya realizo su firma. Existen casos en donde los pacientes abandonan la atención así que el médico tendrá la funcionalidad de registrar abandono del paciente para liberar la atención, además podrá exportar sus reportes en formato Excel y el reporte de su atención de emergencia en formato pdf.

- ✓ Ingresar al sistema.
- ✓ Listar Orden Atención.
- ✓ Gestionar Enfermedad Actual.
- ✓ Gestionar Funciones Vitales.
- ✓ Gestionar Diagnostico.
- ✓ Gestionar Exámenes de Apoyo al Diagnóstico.
- ✓ Gestionar Receta.

- ✓ Gestionar Descanso Médico.
- ✓ Actualizar Alta Médica.
- ✓ Actualizar Abandono.
- ✓ Generar Reporte Nivel Percentil de Cumplimiento.
- ✓ Generar Reporte Emergencia.
- ✓ Generar Reporte de Atención Emergencia.

Perfil 4: Enfermera:

Es el usuario encargado de registrar las notas de enfermería, además visualiza sus atenciones en la bandeja de emergencia en todos los estados (pendiente, en atención y atendido), este usuario realizara el registro de las notas de enfermería y registro de acto enfermería. Existen casos en donde los pacientes abandonan la atención así que la enfermera tendrá la funcionalidad de registrar abandono del paciente para liberar la atención

- ✓ Ingresar al sistema.
- ✓ Listar Orden Atención.
- ✓ Gestionar Notas de Enfermería.
- ✓ Actualizar Acto de Enfermería.
- ✓ Registrar Abandono.

1.1. Especificación de escenarios

Perfil 1: Admisionistas.

- ✓ **Ingresar al sistema.**

Tabla 18 Ingresar al Sistema

ENTRADA	PROCESO	SALIDA
Ingresar usuario, contraseña y tipo usuario	Verificar que el usuario, contraseña y tipo usuario sean los correctos	Pantalla principal de admisionista

Fuente: Elaboración propia

En la tabla N° 18 se puede ver el escenario de ingresar al sistema en donde los datos de entrada son el usuario, contraseña y tipo usuario en este caso es admisionista, luego los datos son procesados a fin de validar que los datos son los correctos, para finalmente obtener la pantalla principal de admisionista.

✓ **Gestionar Usuario.**

En las tablas N° 19 y N°20 se puede ver las entradas, el proceso y la salida de las tareas para el escenario de gestionar usuario que son las siguientes:

Tabla 19 Registrar Usuario

ENTRADA	PROCESO	SALIDA
Ingresar datos necesarios para registrar usuario	Validar datos registrados sean los correctos	Usuario registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 19 se puede ver que se necesita como datos de entrada todos los datos de usuario, luego son procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de usuario registrado satisfactoriamente.

Tabla 20 Actualizar Usuario

ENTRADA	PROCESO	SALIDA
Realizar la búsqueda del usuario ingresando su DNI y así actualizar sus datos	Validar datos actualizados del usuario	Usuario actualizado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 20 se puede ver como dato de entrada la búsqueda de Dni del usuario a actualizar sus datos, luego se actualiza los datos necesarios para ser procesados, para finalmente obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de usuario actualizado satisfactoriamente.

✓ **Gestionar Paciente.**

En las tablas N° 21 y N° 22 podemos observar las entradas, el proceso y la salida de las tareas para el escenario de gestionar paciente que son las siguientes:

Tabla 21 Registrar Paciente

ENTRADA	PROCESO	SALIDA
---------	---------	--------

Ingresar datos necesarios para registrar paciente	Validar datos registrados sean los correctos	Paciente registrado satisfactoriamente
---	--	--

Fuente: Elaboración propia

En la tabla N° 21 se puede ver que se necesita como datos de entrada todos los datos de paciente, luego son procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de paciente registrado satisfactoriamente.

Tabla 22 Actualizar Paciente

ENTRADA	PROCESO	SALIDA
Realizar la búsqueda del paciente ingresando su DNI y así actualizar sus datos	Validar datos actualizados del paciente	Paciente actualizado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 22 se puede ver como dato de entrada la búsqueda de Dni del paciente a actualizar sus datos, luego se actualiza los datos necesarios para ser procesados, para finalmente obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de paciente actualizado satisfactoriamente.

✓ **Crear Orden Atención.**

Tabla 23 Crear Orden Atención

ENTRADA	PROCESO	SALIDA
Realizar la búsqueda del paciente ingresando su DNI para poder crear la orden de atención	Validar datos creados sean los correctos	Orden atención creada satisfactoriamente.

Fuente: Elaboración propia

En la tabla N° 23 se puede ver como dato de entrada la búsqueda de Dni del paciente a crear su orden atención, luego se valida los datos necesarios para ser procesados, para finalmente obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de orden atención creada satisfactoriamente

✓ **Listar Orden Atención.**

Tabla 24 Listar Orden Atención

ENTRADA	PROCESO	SALIDA
Carga del listado de orden de atención generadas en la bandeja admisionista	Carga del listado ordenes de atención en la bandeja admisionista en estado pendiente o en atención	Bandeja del admisionista cargada de órdenes de atención.

Fuente: Elaboración propia

En la tabla N° 24 se puede ver para el escenario de listar orden atención se necesita haber cargado la bandeja admisionista para luego procesar las ordenes atención en estado pendiente o en atención, finalmente se obtiene la bandeja admisionista cargada con las órdenes atención.

Perfil 2: Internista.

✓ **Ingresar al sistema.**

Tabla 25 Ingresar al Sistema

ENTRADA	PROCESO	SALIDA
Ingresar usuario, contraseña y tipo usuario	Verificar que el usuario, contraseña y tipo usuario sean los correctos	Pantalla principal de internista

Fuente: Elaboración propia

En la tabla N° 25 se puede ver el escenario de ingresar al sistema en donde los datos de entrada son el usuario, contraseña y tipo usuario en este caso es internista, luego los datos son procesados a fin de validar que los datos son los correctos, para finalmente obtener la pantalla principal de internista.

✓ **Listar Orden Atención**

Tabla 26 Listar Orden Atención

ENTRADA	PROCESO	SALIDA
Carga del listado de orden de atención	Carga del listado ordenes de atención en la bandeja	Bandeja internista cargada de órdenes de atención.

generadas en la bandeja internista	internista en estado pendiente o en atención	
------------------------------------	--	--

Fuente: Elaboración propia

En la tabla N° 26 se puede ver para el escenario de listar orden atención se necesita haber cargado la bandeja internista para luego procesar las ordenes atención en estado pendiente o en atención, finalmente se obtiene la bandeja internista cargada con las órdenes atención.

✓ **Listar Motivo Atención**

Tabla 27 Listar Motivo Atención

ENTRADA	PROCESO	SALIDA
Carga el listado de motivo de atención	selección de motivo de atención	Motivo atención seleccionado

Fuente: Elaboración propia

En la tabla N° 27 se puede ver el escenario de listar motivo atención se necesita haber abierto una atención de la bandeja internista, para luego procesar el listado de motivos atención, finalmente se selecciona el motivo atención.

✓ **Registrar triaje.**

Tabla 28 Registrar Triage

ENTRADA	PROCESO	SALIDA
Ingresar datos para el registro de triaje	Validar datos registrados sean los correctos	Triage registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 28 se puede ver el escenario de registrar triaje en donde los datos de entrada son todos los datos del registro de triaje, para luego ser procesados y validados, finalmente se obtiene el mensaje de triaje registrado satisfactoriamente.

✓ **Actualizar Abandono.**

Tabla 29 Actualizar Abandono

ENTRADA	PROCESO	SALIDA
Seleccionar orden atención de la bandeja de internista y clic en botón abandono	Procesar evento del botón abandono	Abandono actualizado satisfactorio

Fuente: Elaboración propia

En la tabla N° 29 se puede ver el escenario de actualizar abandono para cambiar el estado de atención de triaje a abandono en la orden de atención en donde el dato de entrada es la selección de una orden atención en la bandeja internista, luego procesar al dar clic en el botón abandono para obtener el mensaje de abandono actualizado satisfactoriamente.

✓ **Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage.**

Tabla 30 Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage

ENTRADA	PROCESO	SALIDA
Seleccionar el periodo del mes y seleccionar el botón buscar	Procesar el evento del botón buscar para cargar la data en la tabla, luego exportar a un formato Excel	Descargar archivo en formato Excel

Fuente: Elaboración propia

En la tabla N° 30 se puede ver las entradas, el proceso y la salida de la tarea para generar el reporte de nivel porcentual de pacientes atendidos en triaje el cual se podrá buscar por periodo del mes, además exportarlo a un formato Excel en el escenario para generar reporte de nivel porcentual de pacientes atendidos en triaje.

✓ **Generar Reporte Triage.**

Tabla 31 Generar Reporte Triage

ENTRADA	PROCESO	SALIDA
Seleccionar el periodo del mes y seleccionar el botón buscar	Procesar el evento del botón buscar para cargar la data en la tabla, luego exportar a un formato Excel	Descargar archivo en formato Excel

Fuente: Elaboración propia

En la tabla N° 31 se puede ver las entradas, el proceso y la salida de la tarea para generar el reporte triage el cual se podrá buscar por periodo del mes, además exportarlo a un formato Excel en el escenario para generar reporte triage.

Perfil 3: Medico.

✓ Ingresar al sistema.

Tabla 32 Ingresar al Sistema

ENTRADA	PROCESO	SALIDA
Ingresar usuario, contraseña y tipo usuario	Verificar que el usuario, contraseña y tipo usuario sean los correctos	Pantalla principal de médico

Fuente: Elaboración propia

En la tabla N° 32 se puede ver el escenario de ingresar al sistema en donde los datos de entrada son el usuario, contraseña y tipo usuario en este caso es médico, luego los datos son procesados a fin de validar que los datos son los correctos, para finalmente obtener la pantalla principal de médico.

✓ Listar Orden Atención

Tabla 33 Listar Orden Atención

ENTRADA	PROCESO	SALIDA
Carga del listado de orden de atención generadas en la bandeja médico	Carga del listado ordenes de atención en la bandeja médico en estado pendiente, en atención y atendido	Bandeja médico cargada de órdenes de atención.

Fuente: Elaboración propia

En la tabla N° 33 se puede ver para el escenario de listar orden atención se necesita haber cargado la bandeja médica para luego procesar las ordenes atención en estado pendiente, en atención o atendido, finalmente se obtiene la bandeja médica cargada con las órdenes atención.

✓ **Gestionar Enfermedad Actual.**

En las tablas N° 34, N° 35 y N° 36 podemos observar las entradas, el proceso y la salida de las tareas para el escenario de gestionar enfermedad actual que son las siguientes:

Tabla 34 Registrar Enfermedad Actual

ENTRADA	PROCESO	SALIDA
Ingresar datos necesarios para registrar en el formato enfermedad actual	Validar datos registrados sean los correctos	Enfermedad Actual registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 34 se puede ver que se necesita como datos de entrada todos los datos del formato enfermedad actual, luego son procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de enfermedad actual registrado satisfactoriamente.

Tabla 35 Cargar Enfermedad Actual

ENTRADA	PROCESO	SALIDA
Cargar data en el formato enfermedad actual al registrar o actualizar	Carga de datos registrados o actualizados anteriormente del formato enfermedad actual	Carga de datos satisfactoria

Fuente: Elaboración propia

En la tabla N° 35 se puede ver que se necesita como datos de entrada todos los datos del formato enfermedad actual, luego de haber sido registrados o actualizados, para finalmente poder obtener como salida la realización de la acción solicitada y tener la data cargada.

Tabla 36 Actualizar Enfermedad Actual

ENTRADA	PROCESO	SALIDA
Actualizar campos del formato enfermedad actual	Validar datos actualizados del formato enfermedad actual	Enfermedad Actual registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 36 se puede ver que se necesita como datos de entrada todos los datos del formato enfermedad actual, luego son modificados y procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de enfermedad actual registrado satisfactoriamente.

✓ **Gestionar Funciones Vitales.**

En las tablas N° 37, N° 38 y N° 39 podemos observar las entradas, el proceso y la salida de las tareas para el escenario de gestionar funciones vitales que son las siguientes:

Tabla 37 Registrar Funciones Vitales

ENTRADA	PROCESO	SALIDA
Ingresar datos necesarios para registrar en el formato funciones vitales	Validar datos registrados sean los correctos	Funciones Vitales registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 37 se puede ver que se necesita como datos de entrada todos los datos del formato funciones vitales, luego son procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de funciones vitales registrado satisfactoriamente.

Tabla 38 Cargar Funciones vitales

ENTRADA	PROCESO	SALIDA
Cargar data en el formato funciones vitales al registrar o actualizar	Carga de datos registrados o actualizados anteriormente del formato funciones vitales	Carga de datos satisfactoria

Fuente: Elaboración propia

En la tabla N° 38 se puede ver que se necesita como datos de entrada todos los datos del formato funciones vitales, luego de haber sido registrados o actualizados, para finalmente poder obtener como salida la realización de la acción solicitada y tener la data cargada.

Tabla 39 Actualizar Funciones Vitales

ENTRADA	PROCESO	SALIDA
---------	---------	--------

Actualizar campos del formato funciones vitales	Validar datos actualizados del formato funciones vitales	Funciones Vitales registrado satisfactoriamente
---	--	---

Fuente: Elaboración propia

En la tabla N° 39 se puede ver que se necesita como datos de entrada todos los datos del formato funciones vitales, luego son modificados y procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de funciones vitales registrado satisfactoriamente.

✓ **Gestionar Diagnóstico.**

En las tablas N° 40, N° 41, N° 42 y N° 43 podemos observar las entradas, el proceso y la salida de las tareas para el escenario de gestionar diagnóstico que son las siguientes:

Tabla 40 Registrar Diagnóstico

ENTRADA	PROCESO	SALIDA
Ingresar datos necesarios para registrar en el formato de diagnóstico	Validar datos registrados sean los correctos	Diagnóstico registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 40 se puede ver que se necesita como datos de entrada todos los datos del formato diagnóstico, luego son procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de diagnóstico registrado satisfactoriamente.

Tabla 41 Cargar Diagnóstico

ENTRADA	PROCESO	SALIDA
Cargar data en el formato de diagnóstico al registrar, eliminar o actualizar	Carga de datos registrados o actualizados anteriormente del formato diagnóstico	Carga de datos satisfactoria

Fuente: Elaboración propia

En la tabla N° 41 se puede ver que se necesita como datos de entrada todos los datos del formato diagnóstico, luego de haber sido registrados o actualizados, para finalmente poder obtener como salida la realización de la acción solicitada y tener la data cargada.

Tabla 42 Actualizar Diagnóstico

ENTRADA	PROCESO	SALIDA
Actualizar campos del formato diagnóstico	Validar datos actualizados del formato diagnóstico	Diagnóstico registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 42 se puede ver que se necesita como datos de entrada todos los datos del formato diagnóstico, luego son modificados y procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de diagnóstico registrado satisfactoriamente.

Tabla 43 Eliminar Diagnóstico

ENTRADA	PROCESO	SALIDA
Seleccionar el botón de eliminar del formato diagnóstico	Procesar evento de botón eliminar	Diagnóstico eliminado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 43 se puede ver que se necesita como dato de entrada seleccionar el botón eliminar en el formato de diagnóstico, luego procesar el evento del botón eliminar para obtener finalmente el mensaje de diagnóstico eliminado satisfactoriamente.

✓ **Gestionar Exámenes de Apoyo al Diagnóstico.**

En las tablas N° 44, N° 45, N° 46 y N° 47 podemos observar las entradas, el proceso y la salida de las tareas para el escenario de gestionar exámenes de apoyo al diagnóstico que son las siguientes:

Tabla 44 Registrar Exámenes de Apoyo al Diagnóstico

ENTRADA	PROCESO	SALIDA
---------	---------	--------

Ingresar datos necesarios para registrar en el formato de exámenes de apoyo al diagnóstico	Validar datos registrados sean los correctos	Exámenes de apoyo al diagnóstico registrado satisfactoriamente
--	--	--

Fuente: Elaboración propia

En la tabla N° 44 se puede ver que se necesita como datos de entrada todos los datos del formato exámenes de apoyo al diagnóstico, luego son procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de exámenes de apoyo al diagnóstico registrado satisfactoriamente.

Tabla 45 Cargar Exámenes de Apoyo al Diagnóstico

ENTRADA	PROCESO	SALIDA
Cargar data en el formato de exámenes de apoyo al diagnóstico al registrar, eliminar o actualizados	Carga de datos registrados o actualizados anteriormente del formato exámenes de apoyo al diagnóstico	Carga de datos satisfactoria

Fuente: Elaboración propia

En la tabla N° 45 se puede ver que se necesita como datos de entrada todos los datos del formato exámenes de apoyo al diagnóstico, luego de haber sido registrados o actualizados, para finalmente poder obtener como salida la realización de la acción solicitada y tener la data cargada.

Tabla 46 Actualizar Exámenes de Apoyo al Diagnóstico

ENTRADA	PROCESO	SALIDA
Actualizar campos del formato exámenes de apoyo al diagnóstico	Validar datos actualizados del formato exámenes de apoyo al diagnóstico	Exámenes de apoyo al diagnóstico registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 46 se puede ver que se necesita como datos de entrada todos los datos del formato exámenes de apoyo al diagnóstico, luego son modificados y procesados,

para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de exámenes de apoyo al diagnóstico registrado satisfactoriamente.

Tabla 47 Eliminar Exámenes de Apoyo al Diagnóstico

ENTRADA	PROCESO	SALIDA
Seleccionar el botón de eliminar del formato exámenes de apoyo al diagnóstico	Procesar evento de botón eliminar	Exámenes de apoyo al diagnóstico eliminado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 47 se puede ver que se necesita como dato de entrada seleccionar el botón eliminar en el formato de exámenes de apoyo al diagnóstico, luego procesar el evento del botón eliminar para obtener finalmente el mensaje de exámenes de apoyo al diagnóstico eliminado satisfactoriamente.

✓ **Gestionar Receta.**

En las tablas N° 48, N° 49, N° 520y N° 51 podemos observar las entradas, el proceso y la salida de las tareas para el escenario de gestionar receta que son las siguientes:

Tabla 48 Registrar Receta

ENTRADA	PROCESO	SALIDA
Ingresar datos necesarios para registrar en el formato de receta	Validar datos registrados sean los correctos	receta registrada satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 48 se puede ver que se necesita como datos de entrada todos los datos del formato receta, luego son procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de receta registrado satisfactoriamente.

Tabla 49 Cargar Receta

ENTRADA	PROCESO	SALIDA
---------	---------	--------

Cargar data en el formato de receta al registrar, eliminar o actualizar	Carga de datos registrados o actualizados anteriormente del formato receta	Carga de datos satisfactoria
---	--	------------------------------

Fuente: Elaboración propia

En la tabla N° 49 se puede ver que se necesita como datos de entrada todos los datos del formato receta, luego de haber sido registrados o actualizados, para finalmente poder obtener como salida la realización de la acción solicitada y tener la data cargada.

Tabla 50 Actualizar Receta

ENTRADA	PROCESO	SALIDA
Actualizar campos del formato receta	Validar datos actualizados del formato receta	receta registrada satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 50 se puede ver que se necesita como datos de entrada todos los datos del formato receta, luego son modificados y procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de receta registrado satisfactoriamente.

Tabla 51 Eliminar Receta

ENTRADA	PROCESO	SALIDA
Seleccionar el botón de eliminar del formato receta	Procesar evento de botón eliminar	receta eliminada satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 51 se puede ver que se necesita como dato de entrada seleccionar el botón eliminar en el formato de receta, luego procesar el evento del botón eliminar para obtener finalmente el mensaje de receta eliminado satisfactoriamente.

✓ **Gestionar Descanso Médico.**

En las tablas N° 52, N° 53 y N° 54 podemos observar las entradas, el proceso y la salida de las tareas para el escenario de gestionar descanso médico que son las siguientes:

Tabla 52 Registrar Descanso Médico

ENTRADA	PROCESO	SALIDA
Ingresar datos necesarios para registrar en el formato descanso médico	Validar datos registrados sean los correctos	Descanso médico registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 52 se puede ver que se necesita como datos de entrada todos los datos del formato descanso médico, luego son procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de descanso médico registrado satisfactoriamente.

Tabla 53 Cargar Descanso Médico

ENTRADA	PROCESO	SALIDA
Cargar data en el formato descanso médico al registrar o actualizar	Carga de datos registrados o actualizados anteriormente del formato descanso médico	Carga de datos satisfactoria

Fuente: Elaboración propia

En la tabla N° 53 se puede ver que se necesita como datos de entrada todos los datos del formato descanso médico, luego de haber sido registrados o actualizados, para finalmente poder obtener como salida la realización de la acción solicitada y tener la data cargada.

Tabla 54 Actualizar Descanso Médico

ENTRADA	PROCESO	SALIDA
Actualizar campos del formato descanso médico	Validar datos actualizados del formato descanso médico	Descanso médico registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 54 se puede ver que se necesita como datos de entrada todos los datos del formato descanso médico, luego son modificados y procesados, para finalmente

poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de descanso médico registrado satisfactoriamente.

✓ **Actualizar Alta Médica.**

Tabla 55 Actualizar Alta Médica

ENTRADA	PROCESO	SALIDA
Seleccionar orden atención de la bandeja de médico y clic en botón alta médica, luego ingresar contraseña y tipo de alta medica	Procesar evento del botón alta médica	Alta Médica actualizado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 55 se poder ver en el escenario de actualizar alta médica como dato de entrada se selecciona una orden atención de la bandeja del médico y clic en el botón de alta médica, luego procesa el botón de alta médica y finalmente muestra mensaje de alta médica actualizado satisfactoriamente.

✓ **Actualizar Abandono.**

Tabla 56 Actualizar abandono

ENTRADA	PROCESO	SALIDA
Seleccionar orden atención de la bandeja de médico y clic en botón abandono	Procesar evento del botón abandono	Abandono actualizado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 56 se puede ver el escenario de actualizar abandono para cambiar el estado de atención de emergencia a abandono en la orden de atención en donde el dato de entrada es la selección de una orden atención en la bandeja médico, luego procesar al dar clic en el botón abandono para obtener el mensaje de abandono actualizado satisfactoriamente.

✓ **Generar Reporte Nivel Percentil de Cumplimiento.**

Tabla 57 Generar Reporte Nivel Percentil de Cumplimiento

ENTRADA	PROCESO	SALIDA
Seleccionar el periodo del mes y seleccionar el botón buscar	Procesar el evento del botón buscar para cargar la data en la tabla, luego exportar a un formato Excel	Descargar archivo en formato Excel

Fuente: Elaboración propia

En la tabla N° 57 podemos observar las entradas, el proceso y la salida de la tarea para generar el reporte de nivel percentil de cumplimiento el cual se podrá buscar por periodo del mes, además exportarlo a un formato Excel en el escenario para generar reporte de nivel percentil de cumplimiento.

✓ **Generar Reporte Emergencia.**

Tabla 58 Generar Reporte Emergencia

ENTRADA	PROCESO	SALIDA
Seleccionar el periodo del mes y seleccionar el botón buscar	Procesar el evento del botón buscar para cargar la data en la tabla, luego exportar a un formato Excel	Descargar archivo en formato Excel

Fuente: Elaboración propia

En la tabla N° 58 podemos observar las entradas, el proceso y la salida de la tarea para generar el reporte emergencia el cual se podrá buscar por periodo del mes, además exportarlo a un formato Excel en el escenario para generar reporte emergencia.

✓ **Generar Reporte Atención Emergencia.**

Tabla 59 Reporte Atención Emergencia

ENTRADA	PROCESO	SALIDA
Seleccionar orden atención de la bandeja	Procesar evento del botón reporte	Descarga archivo en formato pdf satisfactorio

de médico y clic en botón reporte		
-----------------------------------	--	--

Fuente: Elaboración propia

En la tabla N° 59 podemos observar las entradas, el proceso y la salida de la tarea para generar el reporte de atención de emergencia el cual se podrá descargar en formato pdf, en el escenario para generar reporte atención emergencia.

Perfil 4: Enfermera.

✓ Ingresar al sistema.

Tabla 60 Ingresar al Sistema

ENTRADA	PROCESO	SALIDA
Ingresar usuario, contraseña y tipo usuario	Verificar que el usuario, contraseña y tipo usuario sean los correctos	Pantalla principal de enfermera

Fuente: Elaboración propia

En la tabla N° 60 se puede ver el escenario de ingresar al sistema en donde los datos de entrada son el usuario, contraseña y tipo usuario en este caso es enfermera, luego los datos son procesados a fin de validar que los datos son los correctos, para finalmente obtener la pantalla principal de enfermería.

✓ Listar Orden Atención

Tabla 61 Listar Orden Atención

ENTRADA	PROCESO	SALIDA
Carga del listado de orden de atención generadas en la bandeja enfermera	Carga del listado ordenes de atención en la bandeja enfermera en estado pendiente, en atención y atendido	Bandeja enfermera cargada de órdenes de atención.

Fuente: Elaboración propia

En la tabla N° 61 se puede ver para el escenario de listar orden atención se necesita haber cargado la bandeja enfermería para luego procesar las ordenes atención en estado pendiente, en atención o atendido, finalmente se obtiene la bandeja enfermería cargada con las órdenes atención.

✓ Gestionar Notas Enfermería.

En las tablas N° 62, N° 63 y N° 64 podemos observar las entradas, el proceso y la salida de las tareas para el escenario de gestionar notas enfermería que son las siguientes:

Tabla 62 Registrar Notas Enfermería

ENTRADA	PROCESO	SALIDA
Ingresar datos necesarios para registrar en el formato notas enfermería	Validar datos registrados sean los correctos	Notas enfermería registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 62 se puede ver que se necesita como datos de entrada todos los datos del formato notas enfermería, luego son procesados, para finalmente poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de notas enfermería registrado satisfactoriamente.

Tabla 63 Cargar Notas Enfermería

ENTRADA	PROCESO	SALIDA
Cargar data en el formato notas enfermería al registrar o actualizar	Carga de datos registrados o actualizados anteriormente del formato notas enfermería	Carga de datos satisfactoria

Fuente: Elaboración propia

En la tabla N° 63 se puede ver que se necesita como datos de entrada todos los datos del formato notas enfermería, luego de haber sido registrados o actualizados, para finalmente poder obtener como salida la realización de la acción solicitada y tener la data cargada.

Tabla 64 Actualizar Notas Enfermería

ENTRADA	PROCESO	SALIDA
Actualizar campos del formato notas enfermería	Validar datos actualizados del formato notas enfermería	Notas enfermería registrado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 64 se puede ver que se necesita como datos de entrada todos los datos del formato notas enfermería, luego son modificados y procesados, para finalmente

poder obtener como salida la realización de la acción solicitada y tener como respuesta un mensaje de notas enfermería registrado satisfactoriamente.

✓ **Actualizar Acto Enfermería.**

Tabla 65 Actualizar Acto Enfermería

ENTRADA	PROCESO	SALIDA
Seleccionar orden atención de la bandeja de enfermera y clic en botón acto enfermería	Procesar evento del botón acto enfermería	acto enfermería actualizado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 65 se poder ver en el escenario de actualizar acto enfermería como dato de entrada se selecciona una orden atención de la bandeja enfermería y clic en el botón de acto enfermería, luego procesa el botón de acto enfermería y finalmente muestra mensaje de acto enfermería actualizado satisfactoriamente.

✓ **Actualizar Abandono.**

Tabla 66 Actualizar abandono

ENTRADA	PROCESO	SALIDA
Seleccionar orden atención de la bandeja de enfermera y clic en botón abandono	Procesar evento del botón abandono	Abandono actualizado satisfactoriamente

Fuente: Elaboración propia

En la tabla N° 66 se puede ver el escenario de actualizar abandono para cambiar el estado de atención de enfermería a abandono en la orden de atención en donde el dato de entrada es la selección de una orden atención en la bandeja enfermería, luego procesar al dar clic en el botón abandono para obtener el mensaje de abandono actualizado satisfactoriamente.

Especificación de casos de uso

Los casos de uso que surgen a partir de aquí, serán luego representados mediante la iteración del usuario y el sistema, los cuales proveen de una representación gráfica

concisa de la interacción entre el usuario y el sistema durante la ejecución de alguna tarea.

- **Ingresar al Sistema**

En la tabla N° 67 se muestra la especificación del caso de uso “Ingresar al sistema” donde se presenta la descripción, flujo, eventos, entre otros datos que permiten un mejor entendimiento del caso de uso en mención.

Tabla 67 Especificación caso de uso Ingresar al Sistema

Código del caso de uso	CUS01
Actor(es):	Admisionista, Medico, Internista y Enfermera.
Propósito:	Permitir al usuario poder ingresar al sistema con sus respectivos perfiles.
Precondiciones:	El usuario debe estar registrado en la base de datos para poder iniciar sesión.
Post-condiciones	Se mostrará interfaz principal de acuerdo al perfil de usuario.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
1. Ingresar usuario, contraseña y seleccionar el tipo de usuario al cual pertenece, después dar clic al botón ingresar.	2. Muestra pantalla principal de acuerdo al perfil.
FLUJO ALTERNO	

1. Si no completa todos los campos el sistema mostrara mensaje de alerta indicando los campos que falta completar.
2. Si el usuario, contraseña o el tipo de usuario es el incorrecto el sistema mostrara un mensaje de error indicando que las credenciales son incorrectas.

Fuente: Elaboración propia

- **Gestionar Usuario**

Tabla 68 Especificación caso de uso Gestionar Usuario

Código del caso de uso	CUS02
Actor(es):	Admisionista
Propósito:	Permitir al usuario admisionista poder registrar y actualizar un usuario.
Precondiciones:	Haber ingresado al sistema satisfactoriamente.
Post-condiciones	Se ha registrado o actualizado un usuario satisfactoriamente.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Clic en el menú usuario en la página principal admisionista. 3. Clic en el sub menú gestionar usuario. 5. Seleccionar el tipo de usuario a crear. 	<ol style="list-style-type: none"> 2. Despliega el menú usuario. 4. Muestra pantalla de gestión de usuario. 6. Se habilitan y deshabilitan los campos a registrar dependiendo el tipo de usuario seleccionado.

7. Ingresar los datos y clic en registrar	8. Valida los datos ingresados, guarda en la base de datos y muestra mensaje de “usuario registrado satisfactoriamente” y limpia los campos.
FLUJO ALTERNO	
<p>1. Si el usuario admisionista realiza la acción de actualizar debe buscar al usuario ingresando el dni en la caja de texto y dar clic en buscar, luego edita los datos necesarios y da clic en actualizar.</p> <p>2. Si el usuario admisionista no completa los campos obligatorios el sistema mostrara un mensaje de alerta indicando que campos falta completar.</p> <p>3. Si el usuario admisionista por error intenta ingresar un dni ya creado anteriormente y se encuentra registrado en la base de datos el sistema mostrara un mensaje de alerta indicando que el usuario ya se fue creado.</p>	

Fuente: Elaboración propia

▪ **Gestionar Paciente**

Tabla 69 Especificación caso de uso Gestionar Paciente

Código del caso de uso	CUS03
Actor(es):	Admisionista
Propósito:	Permitir al usuario admisionista poder registrar y actualizar un paciente.
Precondiciones:	Haber ingresado al sistema satisfactoriamente.
Post-condiciones	Se ha registrado o actualizado un paciente satisfactoriamente.

FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Clic en el menú paciente en la página principal admisionista. 3. Clic en el sub menú gestionar paciente. 5. Ingresar los datos y clic en registrar 	<ol style="list-style-type: none"> 2. Despliega el menú paciente. 4. Muestra pantalla de gestión de paciente. 6. Valida los datos ingresados, guarda en la base de datos y muestra mensaje de “paciente registrado satisfactoriamente” y limpia los campos.
FLUJO ALTERNO	
<ol style="list-style-type: none"> 1. Si el usuario admisionista realiza la acción de actualizar debe buscar al paciente ingresando el dni en la caja de texto y dar clic en buscar, luego edita los datos necesarios y da clic en actualizar. 2. Si el usuario admisionista no completa los campos obligatorios el sistema mostrara un mensaje de alerta indicando que campos falta completar. 3. Si el usuario admisionista por error intenta ingresar un dni ya creado anteriormente y se encuentra registrado en la base de datos el sistema mostrara un mensaje de alerta indicando que el paciente ya se fue creado. 	

Fuente: Elaboración propia

▪ **Crear Orden Atención**

Tabla 70 Especificación caso de uso Crear Orden Atención

Código del caso de uso	CUS04
Actor(es):	Admisionista

Propósito:	Permitir al usuario admisionista poder crear una orden atención.
Precondiciones:	Haber ingresado al sistema satisfactoriamente.
Post-condiciones	Se ha creado orden atención satisfactoriamente.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Clic en Menú Orden Atención en la página principal admisionista. 3. Clic en Sub Menú Crear Orden Atención. 5. Ingresar dni de paciente y clic en buscar. 7. Selecciona forma de llegada de paciente en el combobox e ingresa el detalle de ingreso en el text area y clic en crear orden atención. 	<ol style="list-style-type: none"> 2. Despliega el menú Orden Atención. 4. Muestra pantalla de Crear Orden Atención. 6. Carga datos de paciente en los campos de la pantalla. 8. Valida los datos ingresados, guarda en la base de datos y muestra mensaje de “orden atención creado satisfactoriamente” y limpia los campos.
FLUJO ALTERNO	
<ol style="list-style-type: none"> 1. Si no completa los datos llegada de paciente y detalle de llegada el sistema mostrara un mensaje de alerta indicando que se debe completar los campos mencionados anteriormente. 2. Si ingresa un dni incorrecto y da clic en buscar el sistema mostrara un mensaje de error indicando que el dni no existe. 	

Fuente: Elaboración propia

- **Listar Orden Atención**

Tabla 71 Especificación caso de uso Listar Orden Atención

Código del caso de uso	CUS05
Actor(es):	Admisionista, Medico, Internista y Enfermera.
Propósito:	Permitir a los usuarios listar las ordenes atención en sus respectivas bandejas.
Precondiciones:	Haber ingresado al sistema satisfactoriamente.
Post-condiciones	Se mostrará interfaz con las ordenes atención cargadas en la bandeja dependiendo cada perfil.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
1. Clic Menú Orden Atención en la página principal de perfil de usuario. 3. Clic Sub Menú Bandeja orden atención.	2. Despliega Menú Orden Atención. 4. Muestra pantalla de bandeja dependiendo el perfil y con el listado cargado.
FLUJO ALTERNO	
1. Si busca puede utilizar los campos de fecha y clic en el botón buscar.	

2. En las bandejas de internista, médico y enfermera si apertura una atención esta se puede realizar dando clic al botón abrir atención.
3. Los listados en las bandejas de médico y enfermera cargaran las ordenes de atención en estado pendiente, en atención y atendido, pero para los listados de las bandejas de admisionista e internista cargaran las ordenes en estado pendiente y en atención.
4. Los listados en las bandejas de médico y enfermera estarán ordenados por la prioridad de atención, mientras que los listados de admisionista e internista estarán ordenados por hora de ingreso.

Fuente: Elaboración propia

▪ **Listar Motivo Atención**

Tabla 72 Especificación caso de uso Listar Motivo Atención

Código del caso de uso	CUS06
Actor(es):	Internista.
Propósito:	Permitir al usuario internista listar los motivos de atención en triaje.
Precondiciones:	Haber ingresado al sistema y la bandeja de internista este cargado de órdenes de atención.
Post-condiciones	Se mostrará interfaz con el listado de motivos atención.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema

1. Seleccionar fila de la bandeja internista en la página principal internista y clic en abrir atención.	2. Muestra pantalla con los motivos de atención cargados.
FLUJO ALTERNO	
1. Clic en cualquier motivo de atención.	

Fuente: Elaboración propia

▪ **Registrar Triage**

Tabla 73 Especificación caso de uso Registrar Triage

Código del caso de uso	CUS07
Actor(es):	Internista.
Propósito:	Permitir al usuario internista registrar su atención de triaje.
Precondiciones:	Haber ingresado al sistema, después haber aperturado una orden de atención del listado de la bandeja internista, luego haber seleccionado un motivo atención.
Post-condiciones	Se mostrará interfaz del registro de triaje.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema

<ol style="list-style-type: none"> 1. Seleccionar discriminador de acuerdo a evaluación. 3. Ingresar datos y clic en registrar 	<ol style="list-style-type: none"> 2. Carga prioridad de atención en el combobox. 4. Valida los datos ingresados, guarda en la base de datos y muestra mensaje de “triaje registrado satisfactoriamente” y muestra pantalla principal de listado de orden atención en la bandeja internista.
FLUJO ALTERNO	
<ol style="list-style-type: none"> 1. Si el usuario internista olvida registrar algún campo obligatorio el sistema mostrara un mensaje de alerta indicando el campo que falta registrar. 2. Al seleccionar un discriminador de un panel automáticamente ocultara los demás paneles y para habilitarlos debe desmarcar las opciones seleccionadas. 3. El usuario internista puede cambiar la prioridad de atención a pesar de que ya haya seleccionado el discriminador de otra prioridad, debido a criterio médico. 	

Fuente: Elaboración propia

▪ **Actualizar Abandono**

Tabla 74 Especificación caso de uso Actualizar Abandono

Código del caso de uso	CUS08
Actor(es):	Internista, Médico y Enfermera.
Propósito:	Permitir al usuario actualizar el estado de pendiente a abandono a una orden atención.

Precondiciones:	Haber ingresado al sistema y sus respectivas bandejas tener cargado órdenes de atención.
Post-condiciones	Se mostrará interfaz con las ordenes atención cargadas en la bandeja dependiendo cada perfil.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
1. Seleccionar fila de su respectiva bandeja de usuario y clic en abandono.	2. Muestra pantalla de bandeja dependiendo el perfil y con el listado cargado, además se muestra mensaje de “abandonado actualizado satisfactoriamente”.
FLUJO ALTERNO	
1. Listar bandeja de orden atención dependiendo cada perfil.	

Fuente: Elaboración propia

- **Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage**

Tabla 75 Especificación caso de uso Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage

Código del caso de uso	CUS09
Actor(es):	Internista.
Propósito:	Permitir al usuario internista poder generar un reporte sobre el indicador

	nivel porcentual de pacientes atendidos en triage y exportarlo en Excel.
Precondiciones:	Haber ingresado al sistema satisfactoriamente.
Post-condiciones	Se mostrará interfaz con la data mensual del nivel porcentual de pacientes atendidos en triage cargada en una bandeja y podrá exportarla en formato Excel.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Clic Menú Reportes en la página principal internista. 3. Clic Sub Menú Reporte Nivel Porcentual de Pacientes Atendidos en Triage. 5. Seleccionar periodo y clic en buscar 7. Clic en el botón Excel 	<ol style="list-style-type: none"> 2. Despliega Menú Reportes. 4. Muestra pantalla de Reporte Nivel Porcentual de Pacientes Atendidos en Triage. 6. Muestra la bandeja cargada con la data del periodo seleccionado. 8. Se descargará reporte en formato Excel.
FLUJO ALTERNO	
<ol style="list-style-type: none"> 1. El usuario puede seleccionar cualquier periodo del listado. 	

Fuente: Elaboración propia

▪ **Generar Reporte Triage**

Tabla 76 Especificación caso de uso Generar Reporte Triage

Código del caso de uso	CUS10
Actor(es):	Internista.
Propósito:	Permitir al usuario internista poder generar un reporte mensual de las atenciones en triage y exportarlo en Excel.
Precondiciones:	Haber ingresado al sistema satisfactoriamente.
Post-condiciones	Se mostrará interfaz con la data mensual de las atenciones en triage y podrá exportarla en formato Excel.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Clic Menú Reportes en la página principal internista. 3. Clic Sub Menú Reporte Triage. 5. Seleccionar periodo y clic en buscar 7. Clic en el botón Excel 	<ol style="list-style-type: none"> 2. Despliega Menú Reportes. 4. Muestra pantalla de Reporte Triage. 6. Muestra la bandeja cargada con la data del periodo seleccionado. 8. Se descargará reporte en formato Excel.
FLUJO ALTERNO	
<ol style="list-style-type: none"> 1. El usuario puede seleccionar cualquier periodo del listado. 	

Fuente: Elaboración propia

▪ Gestionar Enfermedad Actual

**Tabla 77 Especificación caso de uso Gestionar
Enfermedad Actual**

Código del caso de uso	CUS11
Actor(es):	Médico
Propósito:	Permitir al usuario médico poder registrar, actualizar y cargar data en el formato de enfermedad actual.
Precondiciones:	Haber ingresado al sistema satisfactoriamente y la bandeja del médico tener listado las ordenes atención.
Post-condiciones	Se registró enfermedad actual satisfactoriamente.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
1. En el listado orden atención en la bandeja del médico seleccionar una orden atención y clic en abrir atención. 3. Clic en Enfermedad actual. 5. Ingresar datos en los campos correspondientes y clic en registrar.	2. Muestra pantalla de atención emergencia. 4. Muestra pantalla de formato Enfermedad Actual. 6. Valida los datos ingresados, registra en la base de datos y muestra mensaje de “enfermedad actual registrado satisfactoriamente” y carga data en los campos del formato.

FLUJO ALTERNO
<ol style="list-style-type: none"> 1. Si edita, tan solo edita los datos necesarios y clic en registrar. 2. El formato carga data apenas se registra o edita. 3. Valida campos obligatorios indicando un mensaje de alerta si es que no se registrara el campo correspondiente.

Fuente: Elaboración propia

▪ **Gestionar Funciones Vitales**

Tabla 78 Especificación caso de uso Gestionar Funciones Vitales

Código del caso de uso	CUS11
Actor(es):	Médico
Propósito:	Permitir al usuario médico poder registrar, actualizar y cargar data en el formato de funciones vitales.
Precondiciones:	Haber ingresado al sistema satisfactoriamente y la bandeja del médico tener listado las ordenes atención.
Post-condiciones	Se registró funciones vitales satisfactoriamente.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema

<ol style="list-style-type: none"> 1. En el listado orden atención en la bandeja del médico seleccionar una orden atención y clic en abrir atención. 3. Clic en Funciones Vitales. 5. Ingresar datos en los campos correspondientes y clic en registrar. 	<ol style="list-style-type: none"> 2. Muestra pantalla de atención emergencia. 4. Muestra pantalla de formato Funciones vitales. 6. Valida los datos ingresados, registra en la base de datos y muestra mensaje de “funciones vitales registrado satisfactoriamente” y carga data en los campos del formato.
FLUJO ALTERNO	
<ol style="list-style-type: none"> 1. Si edita, tan solo edita los datos necesarios y clic en registrar. 2. El formato carga data apenas se registra o edita. 3. Valida campos obligatorios indicando un mensaje de alerta si es que no se registrara el campo correspondiente. 	

Fuente: Elaboración propia

▪ **Gestionar Diagnóstico**

Tabla 79 Especificación caso de uso Gestionar Diagnóstico

Código del caso de uso	CUS11
Actor(es):	Médico
Propósito:	Permitir al usuario médico poder registrar, actualizar y cargar data en el formato de diagnóstico.

Precondiciones:	Haber ingresado al sistema satisfactoriamente y la bandeja del médico tener listado las ordenes atención.
Post-condiciones	Se registró diagnóstico satisfactoriamente.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. En el listado orden atención en la bandeja del médico seleccionar una orden atención y clic en abrir atención. 3. Clic en Diagnóstico. 5. Clic en agregar Diagnóstico 7. Busca diagnóstico en la caja de texto buscar. 9. Clic en botón de icono agregar. 11. Ingresar datos en los campos correspondientes y clic en registrar. 	<ol style="list-style-type: none"> 2. Muestra pantalla de atención emergencia. 4. Muestra pantalla de formato Diagnóstico. 6. Muestra pantalla modal de listado de diagnósticos. 8. Carga de listado con la opción que está buscando. 10. Se carga diagnóstico seleccionado en la pantalla de diagnóstico. 12. Valida los datos ingresados, guarda en la base de datos y muestra mensaje de “diagnóstico registrado satisfactoriamente” y carga data en los campos del formato.
FLUJO ALTERNO	
<ol style="list-style-type: none"> 1. Si edita, tan solo edita los datos necesarios y clic en registrar. 2. El formato carga data apenas se registra o edita. 3. Valida campos obligatorios indicando un mensaje de alerta si es que no se registrara el campo correspondiente. 	

Fuente: Elaboración propia

▪ Gestionar Exámenes de Apoyo al Diagnóstico

Tabla 80 Especificación caso de uso Gestionar Exámenes de Apoyo al Diagnóstico

Código del caso de uso	CUS11
Actor(es):	Médico
Propósito:	Permitir al usuario médico poder registrar, actualizar y cargar data en el formato de exámenes de apoyo al diagnóstico.
Precondiciones:	Haber ingresado al sistema satisfactoriamente y la bandeja del médico tener listado las ordenes atención.
Post-condiciones	Se registró exámenes de apoyo al diagnóstico satisfactoriamente.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
1. En el listado orden atención en la bandeja del médico seleccionar una orden atención y clic en abrir atención. 3. Clic en Exámenes de Apoyo al Diagnóstico. 5. Clic en agregar examen de apoyo al diagnóstico. 7. Busca examen de apoyo al diagnóstico en la caja de texto buscar. 9. Clic en botón de icono agregar. 11. Ingresar datos en los campos correspondientes y clic en registrar.	2. Muestra pantalla de atención emergencia. 4. Muestra pantalla de formato Exámenes de Apoyo al Diagnóstico. 6. Muestra pantalla modal de listado de exámenes de apoyo al diagnóstico. 8. Carga de listado con la opción que está buscando. 10. Se carga examen de apoyo al diagnóstico seleccionado en la pantalla de Exámenes de Apoyo al Diagnóstico. 12. Valida los datos ingresados, guarda en la base de datos y muestra mensaje de “Examen de apoyo al diagnóstico registrado satisfactoriamente” y carga data en los campos del formato.

FLUJO ALTERNO
<ol style="list-style-type: none"> 1. Si edita, tan solo edita los datos necesarios y clic en registrar. 2. El formato carga data apenas se registra o edita. 3. Valida campos obligatorios indicando un mensaje de alerta si es que no se registrara el campo correspondiente.

Fuente: Elaboración propia

▪ **Gestionar Receta**

Tabla 81 Especificación caso de uso Gestionar Receta

Código del caso de uso	CUS11
Actor(es):	Médico
Propósito:	Permitir al usuario médico poder registrar, actualizar y cargar data en el formato de receta.
Precondiciones:	Haber ingresado al sistema satisfactoriamente y la bandeja del médico tener listado las ordenes atención.
Post-condiciones	Se registró receta satisfactoriamente.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema

<ol style="list-style-type: none"> 1. En el listado orden atención en la bandeja del médico seleccionar una orden atención y clic en abrir atención. 3. Clic en Receta. 5. Clic en agregar medicamento. 7. Busca medicamento en la caja de texto buscar. 9. Se registra datos necesarios y clic en botón de icono agregar. 11. Ingresar datos en los campos correspondientes y clic en registrar. 	<ol style="list-style-type: none"> 2. Muestra pantalla de atención emergencia. 4. Muestra pantalla de formato Receta. 6. Muestra pantalla modal de listado de medicamentos. 8. Carga de listado con la opción que está buscando. 10. Se carga medicamento seleccionado en la pantalla de Receta. 12. Valida los datos ingresados, guarda en la base de datos y muestra mensaje de “Receta registrado satisfactoriamente” y carga data en los campos del formato.
FLUJO ALTERNO	
<ol style="list-style-type: none"> 1. Si edita, tan solo edita los datos necesarios y clic en registrar. 2. El formato carga data apenas se registra o edita. 3. Valida campos obligatorios indicando un mensaje de alerta si es que no se registrara el campo correspondiente. 	

Fuente: Elaboración propia

▪ **Gestionar Descanso Médico**

Tabla 82 Especificación caso de uso Gestionar Descanso Médico

Código del caso de uso	CUS11
Actor(es):	Médico
Propósito:	Permitir al usuario médico poder registrar, actualizar y cargar data en el formato de descanso médico.

Precondiciones:	Haber ingresado al sistema satisfactoriamente y la bandeja del médico tener listado las ordenes atención.
Post-condiciones	Se registró descanso médico satisfactoriamente.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. En el listado orden atención en la bandeja del médico seleccionar una orden atención y clic en abrir atención. 3. Clic en Descanso Médico. 5. Ingresar datos en los campos correspondientes y clic en registrar. 	<ol style="list-style-type: none"> 2. Muestra pantalla de atención emergencia. 4. Muestra pantalla de formato Descanso Médico. 6. Valida los datos ingresados, registra en la base de datos y muestra mensaje de “Descanso Médico registrado satisfactoriamente” y carga data en los campos del formato.
FLUJO ALTERNO	
<ol style="list-style-type: none"> 1. Si edita, tan solo edita los datos necesarios y clic en registrar. 2. El formato carga data apenas se registra o edita. 3. Valida campos obligatorios indicando un mensaje de alerta si es que no se registrara el campo correspondiente. 	

Fuente: Elaboración propia

- **Actualizar Alta Médica**

Tabla 83 Especificación caso de uso Actualizar Alta Médica

Código del caso de uso	CUS08
Actor(es):	Médico.
Propósito:	Permitir al usuario médico actualizar el estado de emergencia de en atención a alta médica a una orden atención.
Precondiciones:	Haber registrado diagnóstico de atención y atención de enfermería en estado atendido.
Post-condiciones	Se mostrará mensaje de alta médica satisfactoria y se visualizará el cambio de estado de atendido en la bandeja de médico.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
1. Seleccionar fila de su respectiva bandeja médico y clic en alta médica. 3. Registra los datos necesarios, selecciona el tipo de alta médica y clic en finalizar.	2. Muestra pantalla modal de alta médica. 4. Muestra mensaje de “Alta médica satisfactoria”.

FLUJO ALTERNO
1. Se puede adjuntar un archivo si es que el alta médica es de tipo de defunción o alta voluntaria.

Fuente: Elaboración propia

▪ **Generar Reporte Nivel Percentil de Cumplimiento**

Tabla 84 Especificación caso de uso Generar Reporte Nivel Percentil de Cumplimiento

Código del caso de uso	CUS09
Actor(es):	Médico.
Propósito:	Permitir al usuario médico poder generar un reporte sobre el indicador nivel percentil de cumplimiento y exportarlo en Excel.
Precondiciones:	Haber ingresado al sistema satisfactoriamente.
Post-condiciones	Se mostrará interfaz con la data mensual del nivel percentil de cumplimiento cargada en una bandeja y podrá exportarla en formato Excel.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema

1. Clic Menú Reportes en la página principal médico. 3. Clic Sub Menú Reporte Nivel Percentil de Cumplimiento. 5. Seleccionar periodo y clic en buscar 7. Clic en el botón Excel	2. Despliega Menú Reportes. 4. Muestra pantalla de Reporte Nivel Percentil de Cumplimiento. 6. Muestra la bandeja cargada con la data del periodo seleccionado. 8. Se descargará reporte en formato Excel.
FLUJO ALTERNO	
1. El usuario puede seleccionar cualquier periodo del listado.	

Fuente: Elaboración propia

▪ **Generar Reporte Emergencia**

Tabla 85 Especificación caso de uso Generar Reporte Emergencia

Código del caso de uso	CUS10
Actor(es):	Médico.
Propósito:	Permitir al usuario médico poder generar un reporte mensual de las atenciones en emergencia y exportarlo en Excel.
Precondiciones:	Haber ingresado al sistema satisfactoriamente.
Post-condiciones	Se mostrará interfaz con la data mensual de las atenciones en

	emergencia y podrá exportarla en formato Excel.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
1. Clic Menú Reportes en la página principal médico. 3. Clic Sub Menú Reporte Emergencia. 5. Seleccionar periodo y clic en buscar 7. Clic en el botón Excel	2. Despliega Menú Reportes. 4. Muestra pantalla de Reporte Emergencia. 6. Muestra la bandeja cargada con la data del periodo seleccionado. 8. Se descargará reporte en formato Excel.
FLUJO ALTERNO	
1. El usuario puede seleccionar cualquier periodo del listado.	

Fuente: Elaboración propia

▪ **Generar Reporte de Atención Emergencia**

Tabla 86 Especificación caso de uso Actualizar Alta Médica

Código del caso de uso	CUS08
Actor(es):	Médico.

Propósito:	Permitir al usuario médico poder generar un reporte de la atención y exportarlo en pdf.
Precondiciones:	Haber realizado la atención y esta se encuentre en estado atendido.
Post-condiciones	Se descarga un resumen de la atención en formato pdf.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
1. Seleccionar orden atención de bandeja médico y clic en imprimir.	2. Se descarga formato en pdf.
FLUJO ALTERNO	

Fuente: Elaboración propia

▪ Gestionar Notas Enfermería

Tabla 87 Especificación caso de uso Gestionar Notas Enfermería

Código del caso de uso	CUS11
Actor(es):	Enfermera
Propósito:	Permitir al usuario enfermera poder registrar, actualizar y cargar data en el formato de notas enfermería.
Precondiciones:	Haber ingresado al sistema satisfactoriamente y la bandeja de enfermera tener listado las ordenes atención.
Post-condiciones	Se registró notas enfermería satisfactoriamente.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. En el listado orden atención en la bandeja de enfermera seleccionar una orden atención y clic en abrir atención. 3. Clic en Notas Enfermería. 5. Ingresar datos en los campos correspondientes y clic en registrar. 	<ol style="list-style-type: none"> 2. Muestra pantalla de atención enfermería. 4. Muestra pantalla de formato Notas Enfermería. 6. Valida los datos ingresados, registra en la base de datos y muestra mensaje de “Notas enfermería registrado satisfactoriamente” y carga data en los campos del formato.

FLUJO ALTERNO
<ol style="list-style-type: none"> 1. Si edita, tan solo edita los datos necesarios y clic en registrar. 2. El formato carga data apenas se registra o edita. 3. Valida campos obligatorios indicando un mensaje de alerta si es que no se registrara el campo correspondiente.

Fuente: Elaboración propia

▪ **Actualizar Acto Enfermería**

Tabla 88 Especificación caso de uso Actualizar Acto Enfermería

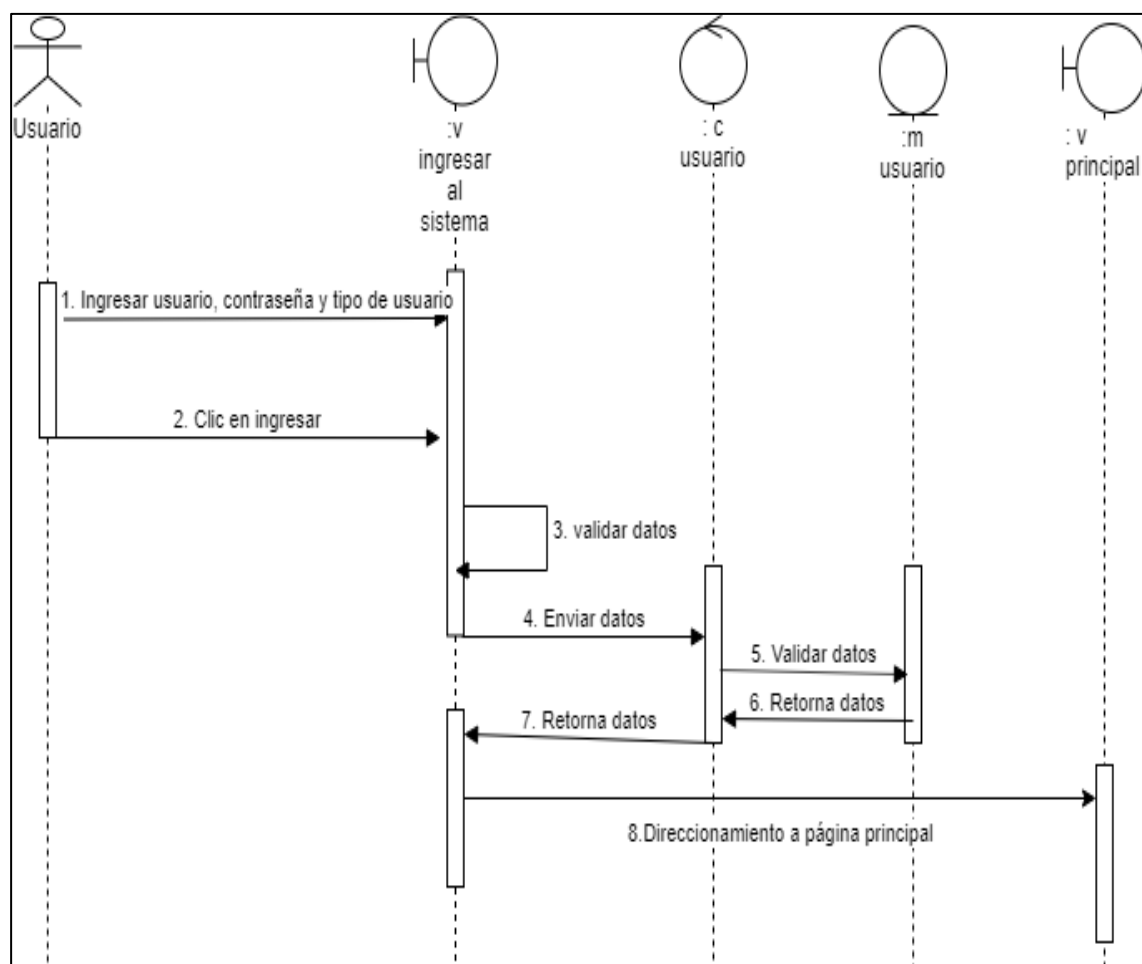
Código del caso de uso	CUS08
Actor(es):	Enfermera.
Propósito:	Permitir al usuario enfermera actualizar el estado de enfermería de en atención a atendido a una orden atención.
Precondiciones:	Haber registrado notas enfermería.
Post-condiciones	Se mostrará mensaje de acto enfermera satisfactoria y se visualizará el cambio de estado de atendido en la bandeja de enfermera.
FLUJO BÁSICO	
Acción del Actor	Respuesta del Sistema

1. Seleccionar fila de su respectiva bandeja enfermera y clic en acto enfermería.	2. Muestra mensaje de “Acto enfermería satisfactoria”.
FLUJO ALTERNO	

Fuente: Elaboración propia

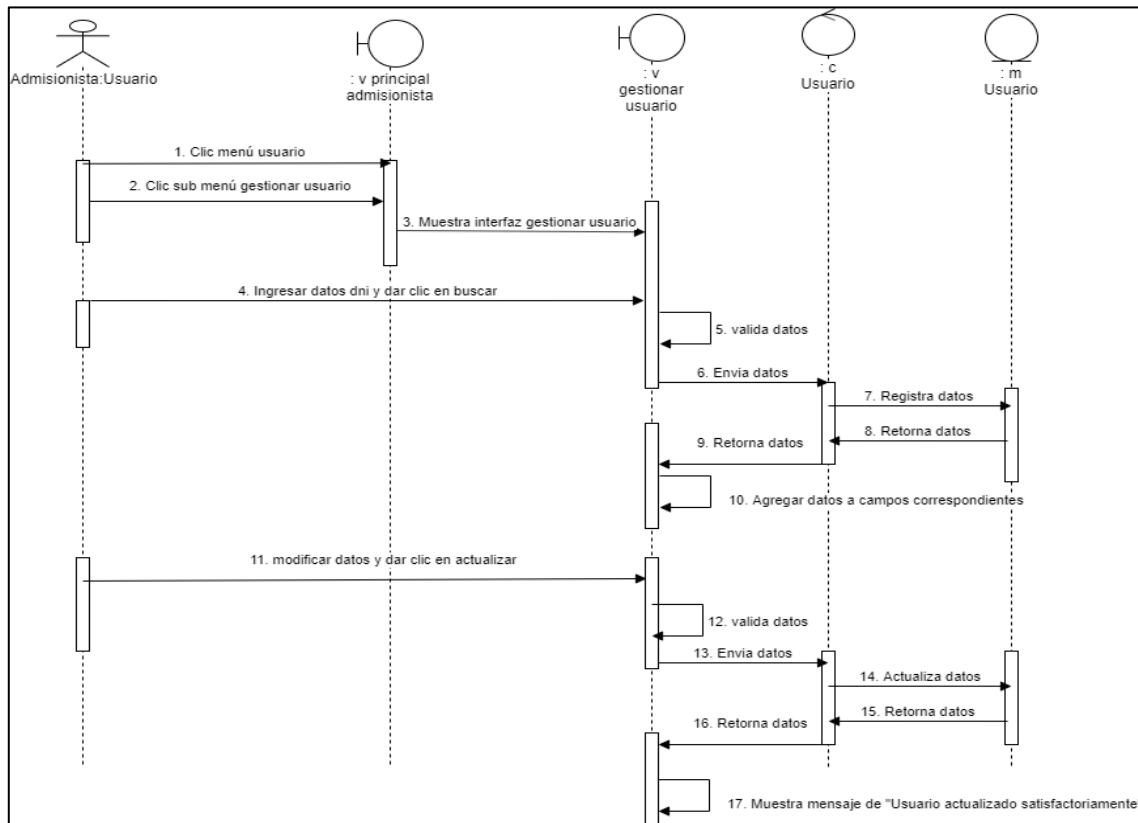
Especificación de caso de diagrama de interacción de usuario

Figura 26 Diagrama de Secuencia Ingresar al Sistema



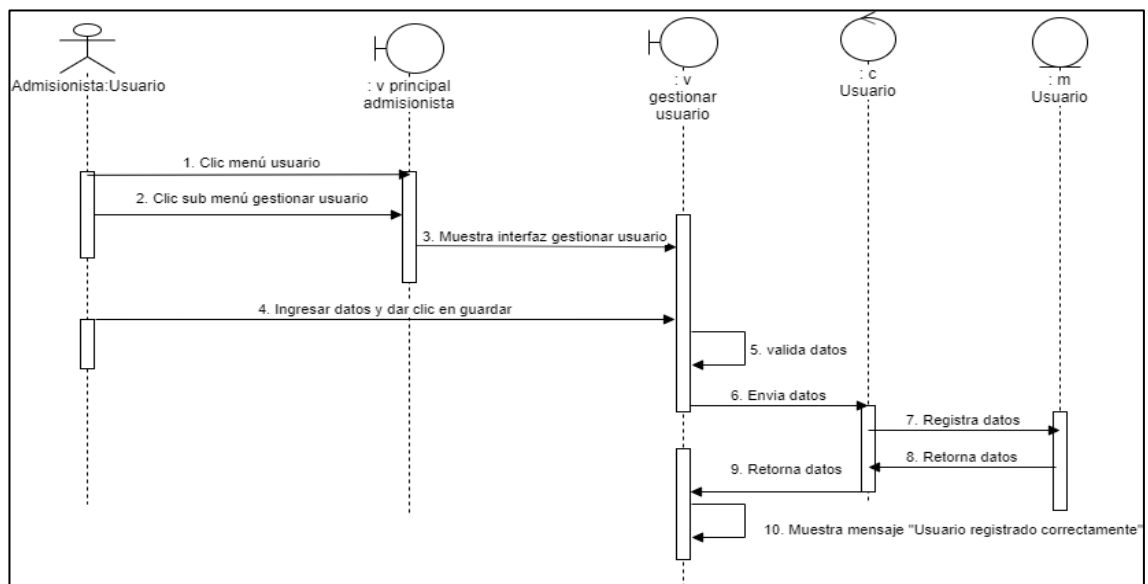
Fuente: Elaboración propia

Figura 27 Diagrama de Secuencia Actualizar Usuario



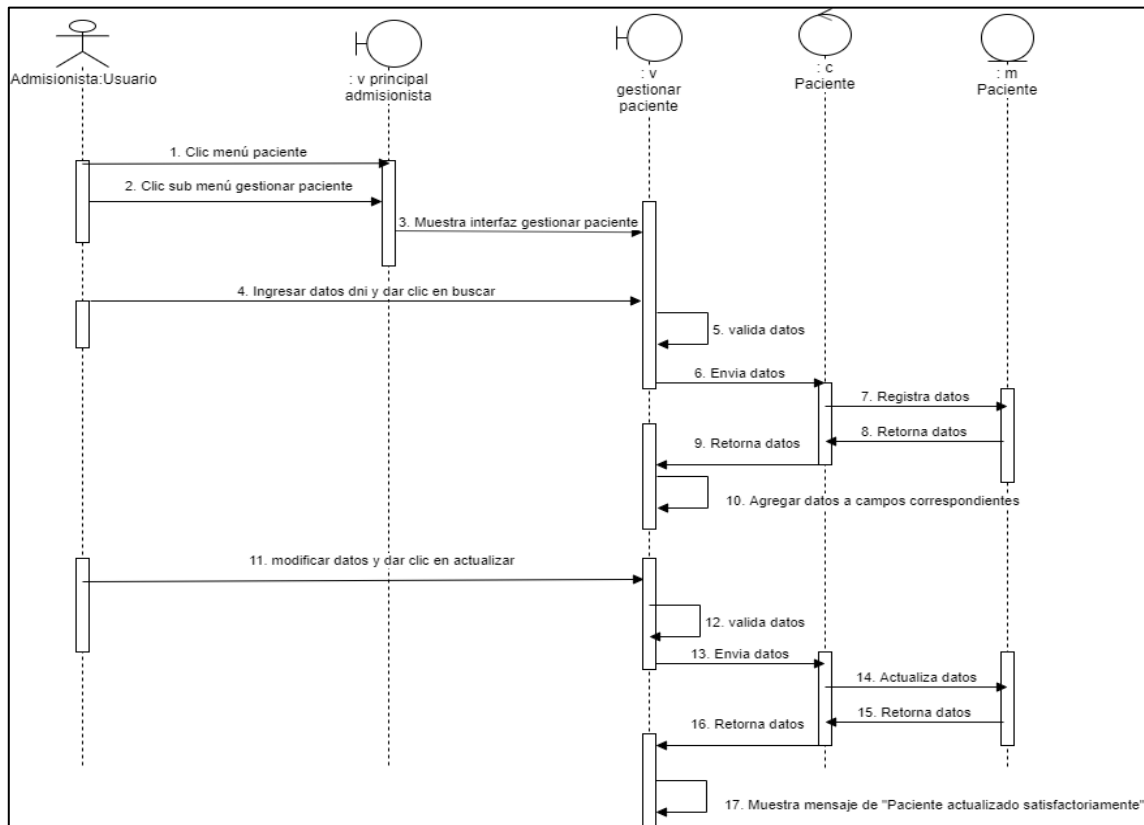
Fuente: Elaboración propia

Figura 28 Diagrama de Secuencia Registrar Usuario



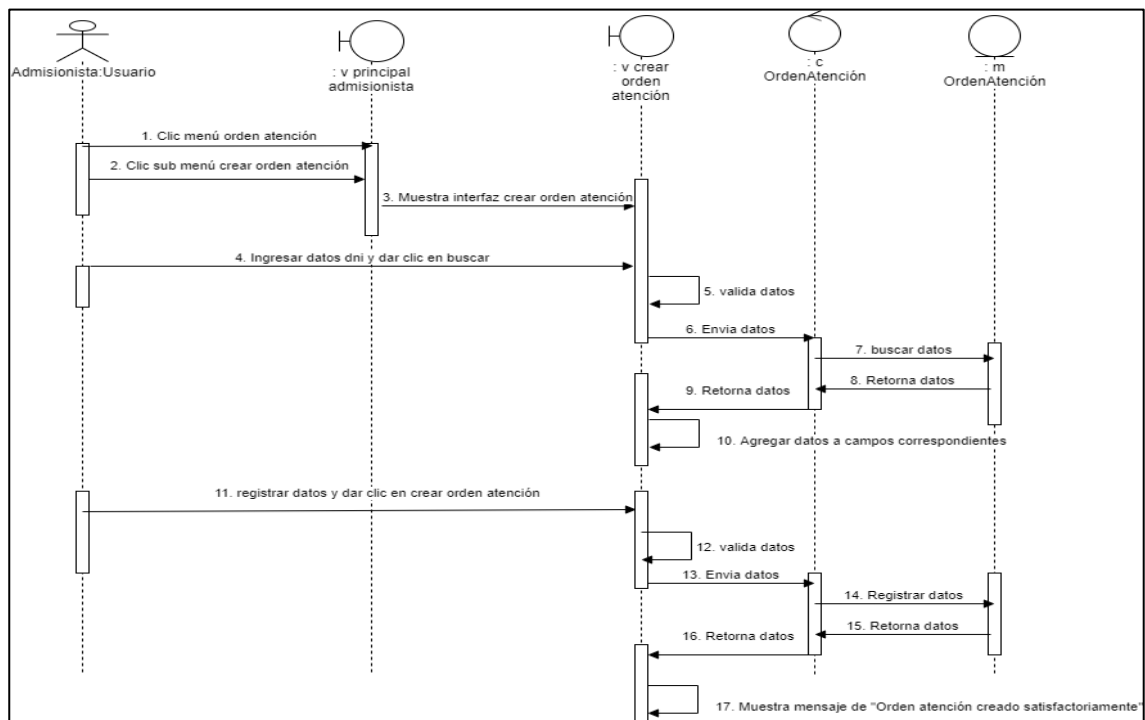
Fuente: Elaboración propia

Figura 29 Diagrama de Secuencia Actualizar Paciente



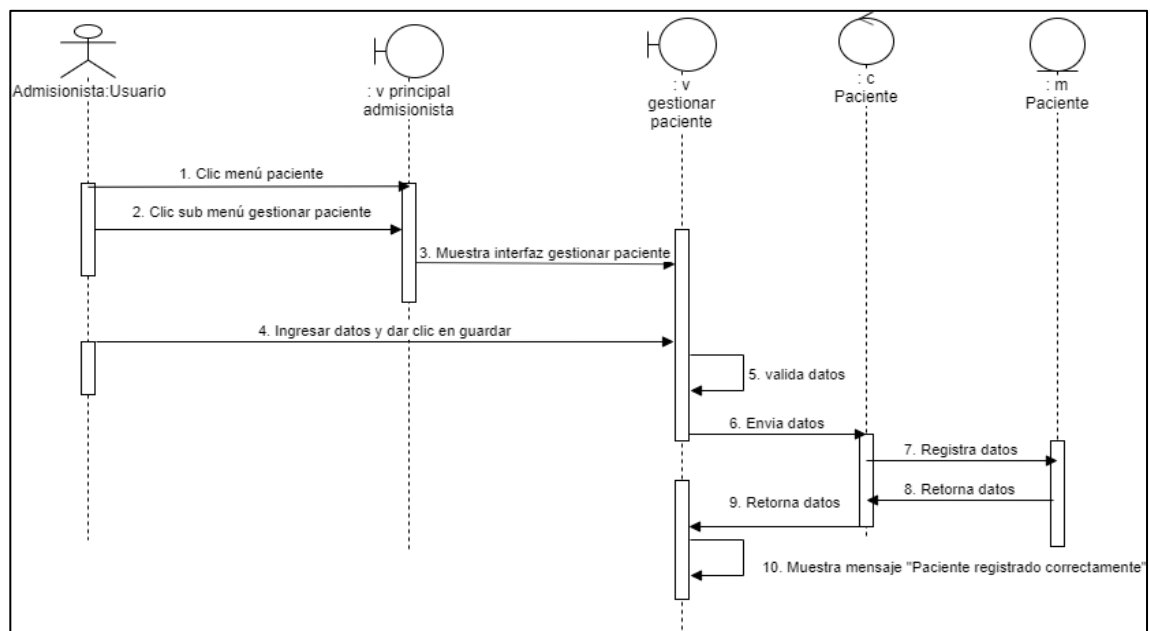
Fuente: Elaboración propia

Figura 30 Diagrama de Secuencia Registrar Paciente



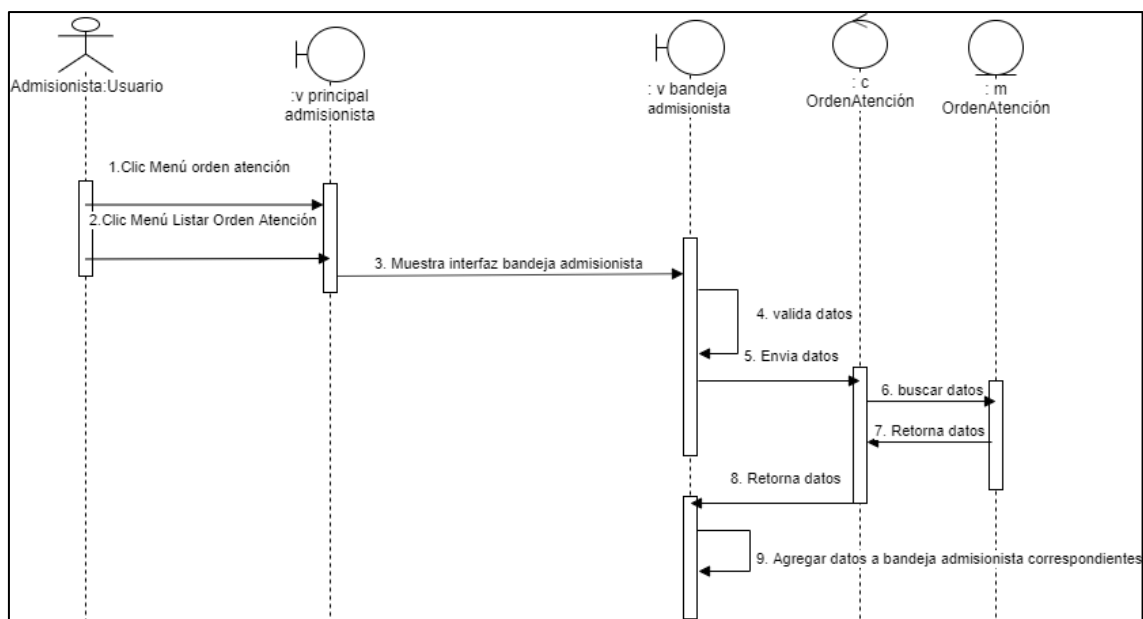
Fuente: Elaboración propia

Figura 31 Diagrama de Secuencia Listar Orden Atención - Admisionista



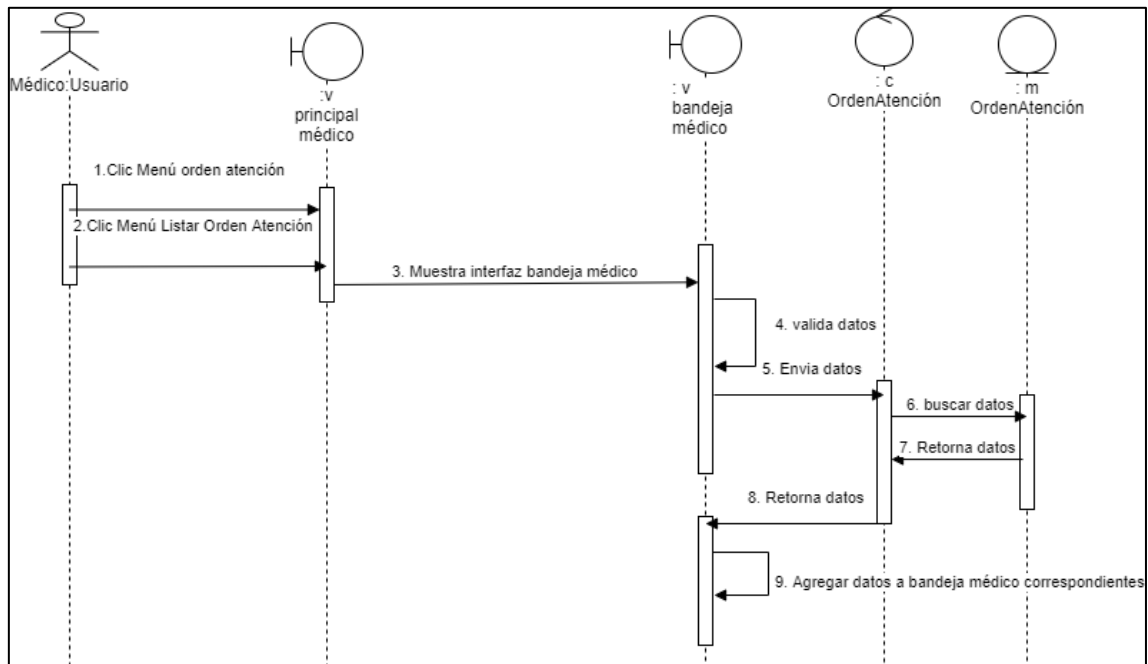
Fuente: Elaboración propia

Figura 32 Diagrama de Secuencia Crear Orden Atención



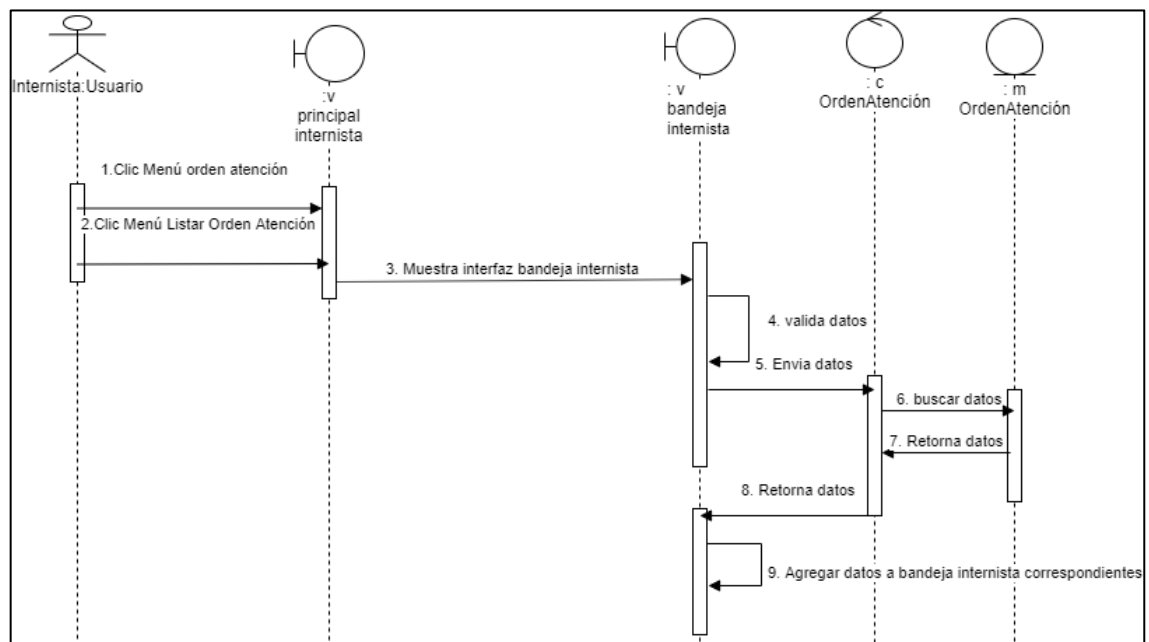
Fuente: Elaboración propia

Figura 33 Diagrama de Secuencia Listar Orden Atención - Internista



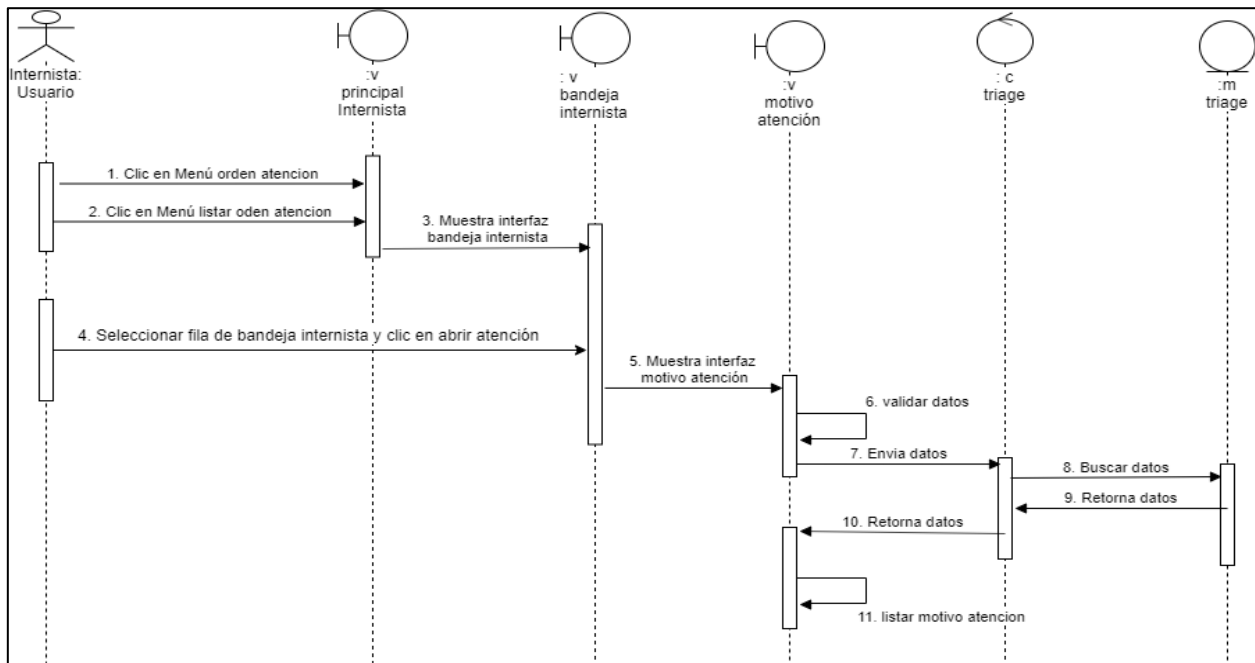
Fuente: Elaboración propia

Figura 34 Diagrama de Secuencia Listar Orden Atención - Médico



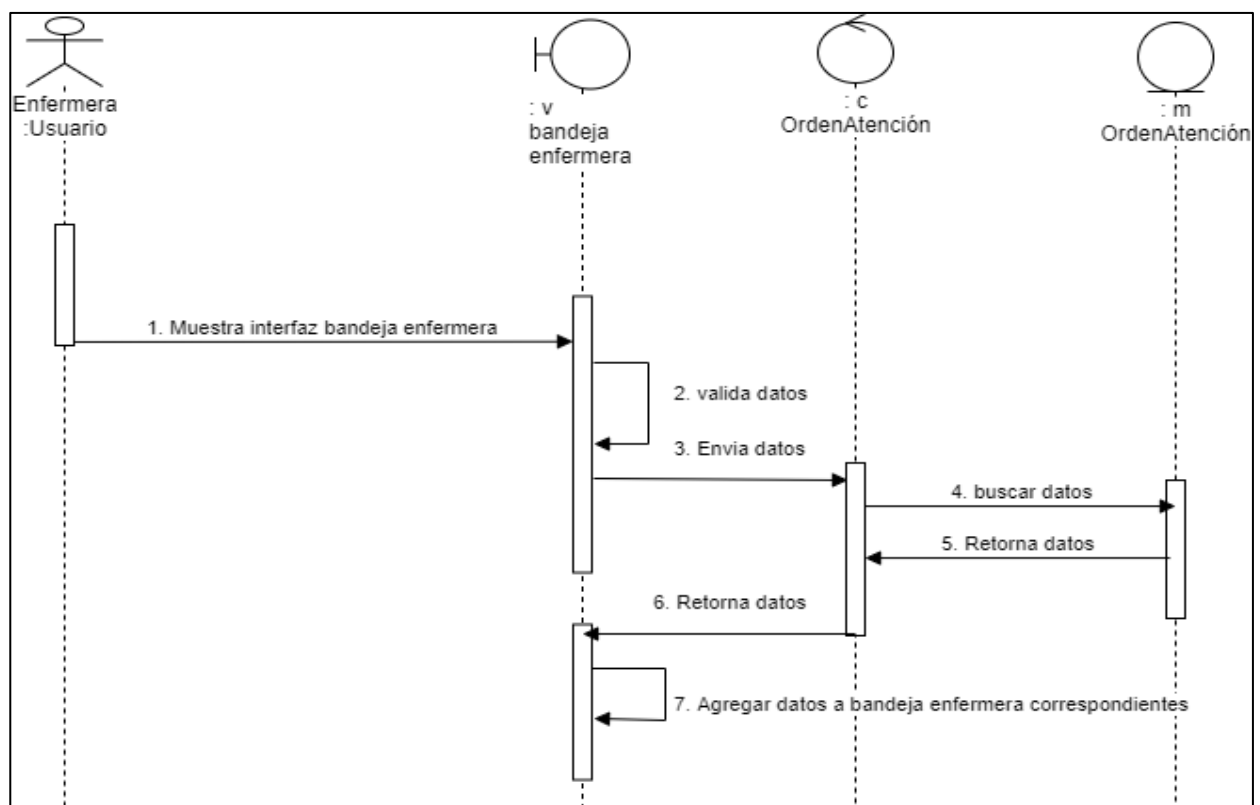
Fuente: Elaboración propia

Figura 36 Diagrama de Secuencia Listar Orden Atención – Enfermera



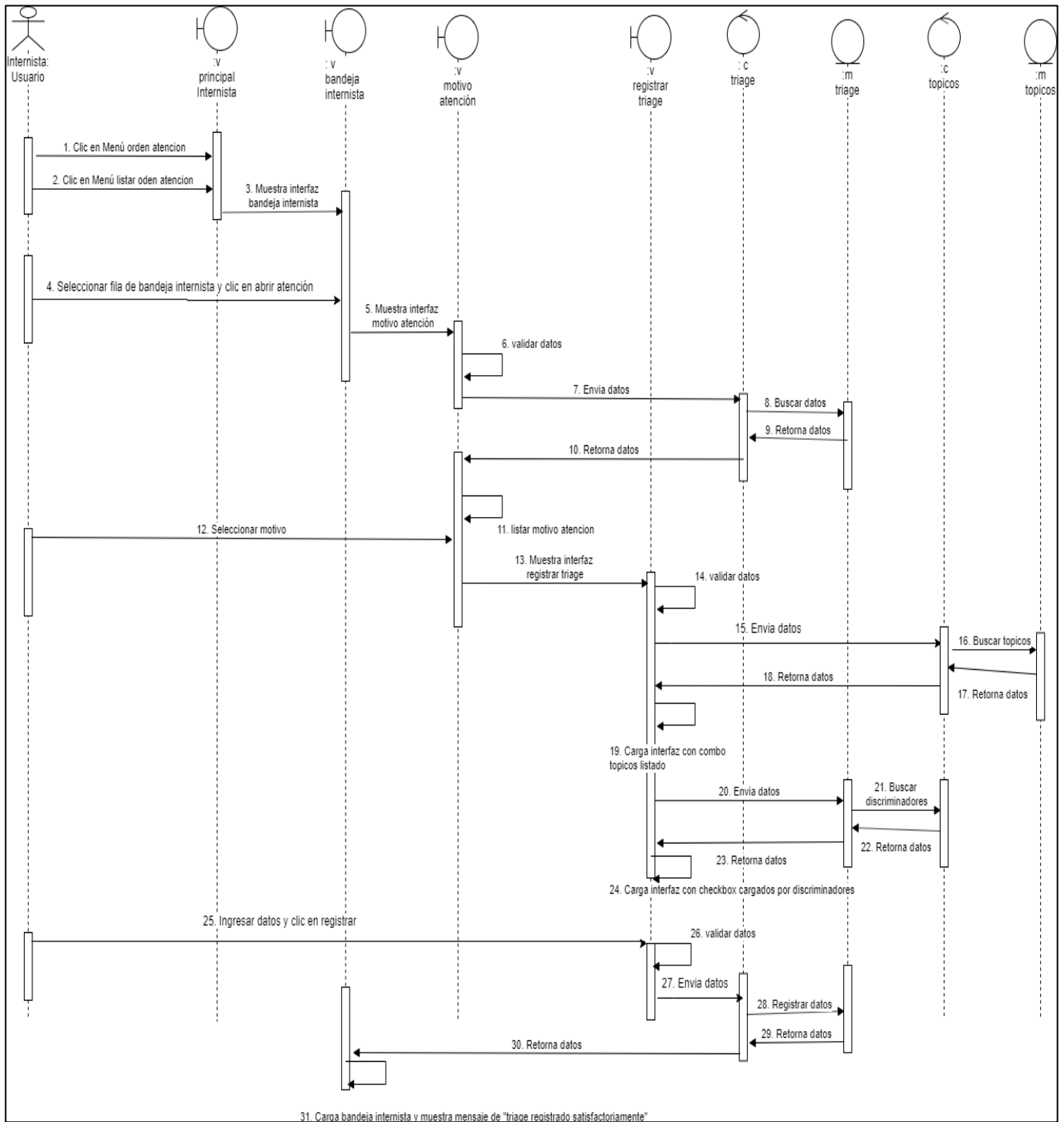
Fuente: Elaboración propia

Figura 35 Diagrama de Secuencia Listar Motivo Atención



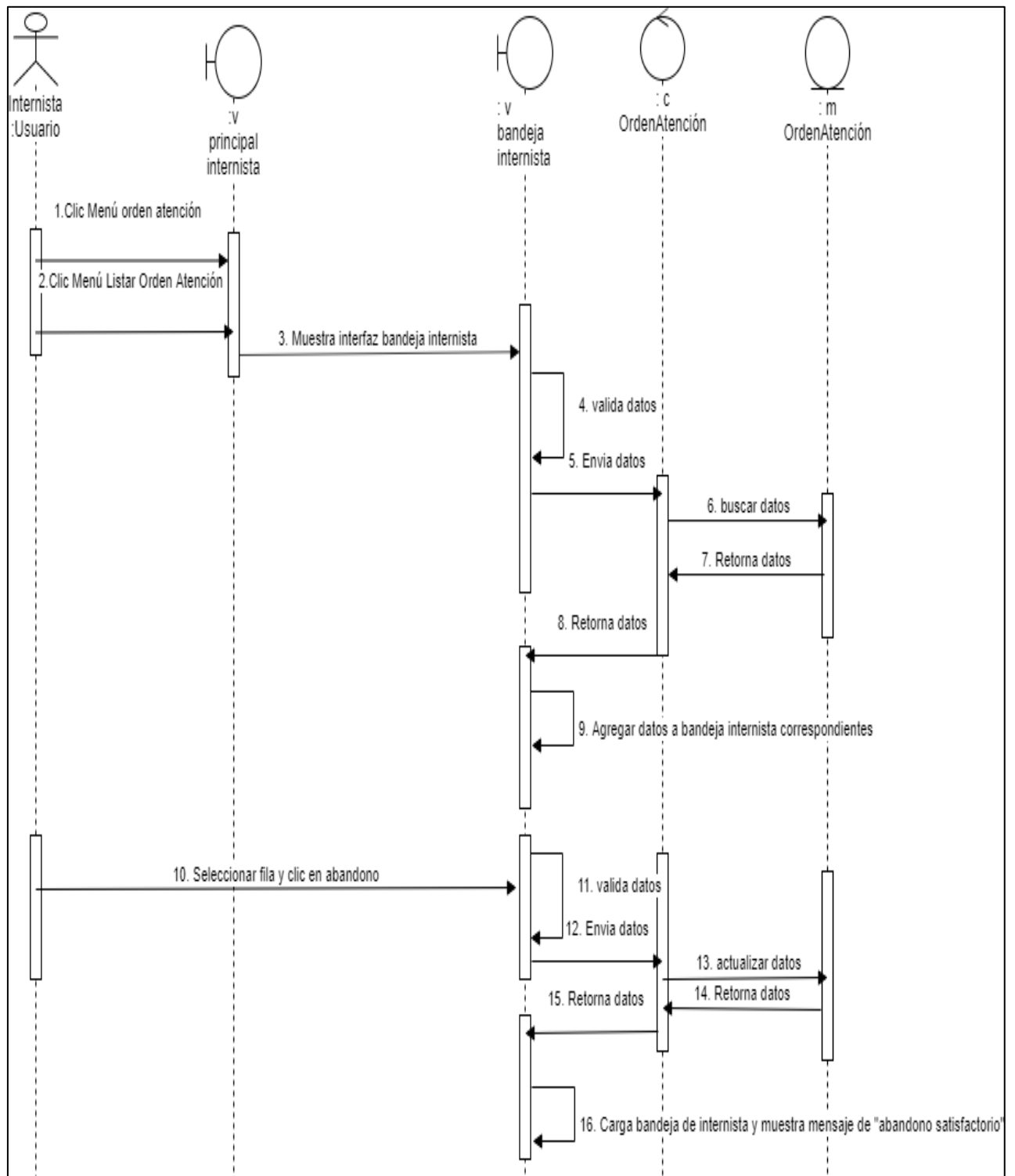
Fuente: Elaboración propia

Figura 37 Diagrama de Secuencia Registrar Triage



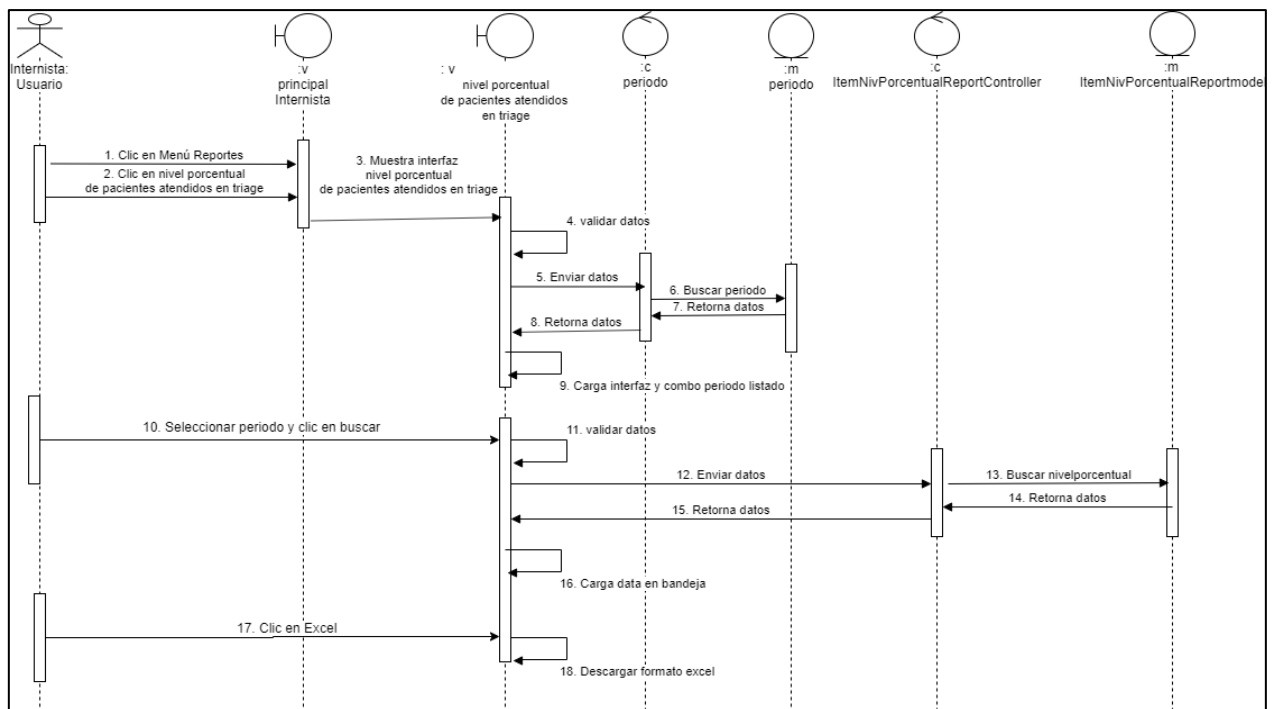
Fuente: Elaboración propia

Figura 38 Diagrama de Secuencia Abandono – Internista



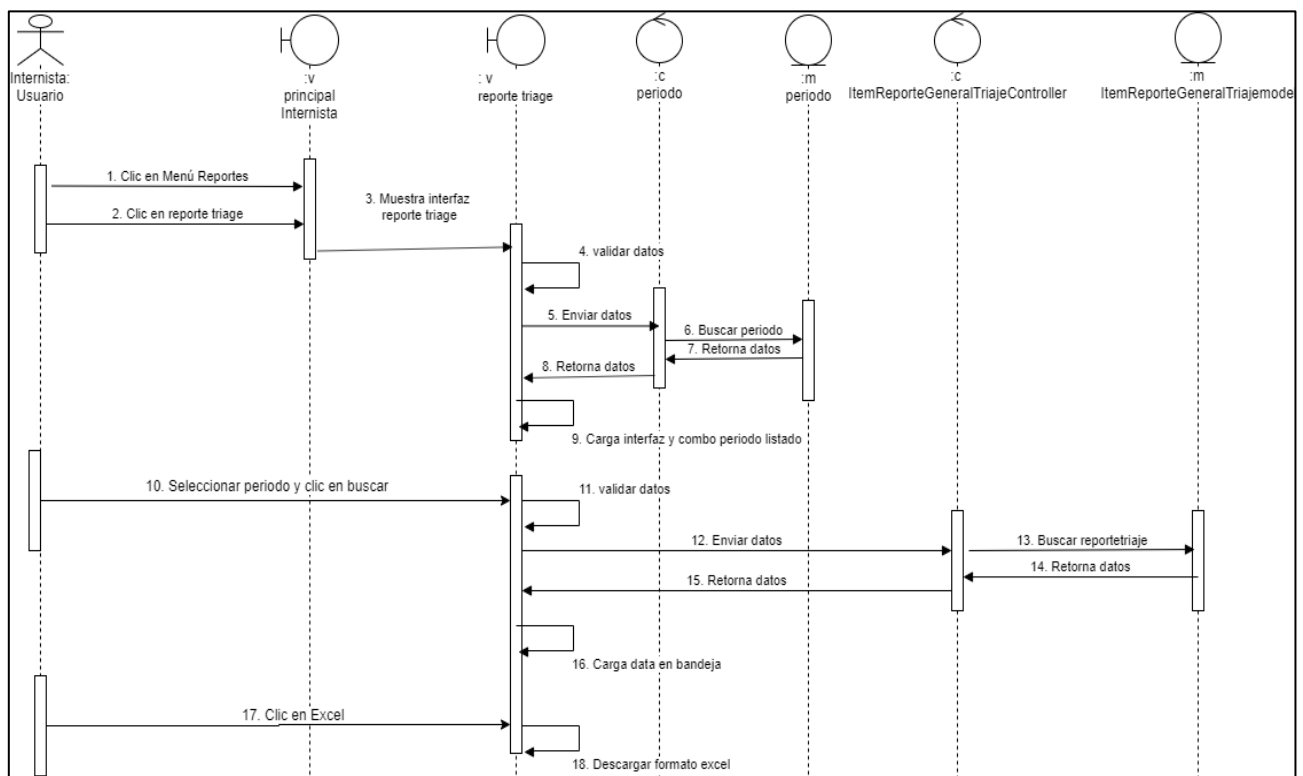
Fuente: Elaboración propia

Figura 39 Diagrama de Secuencia Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage



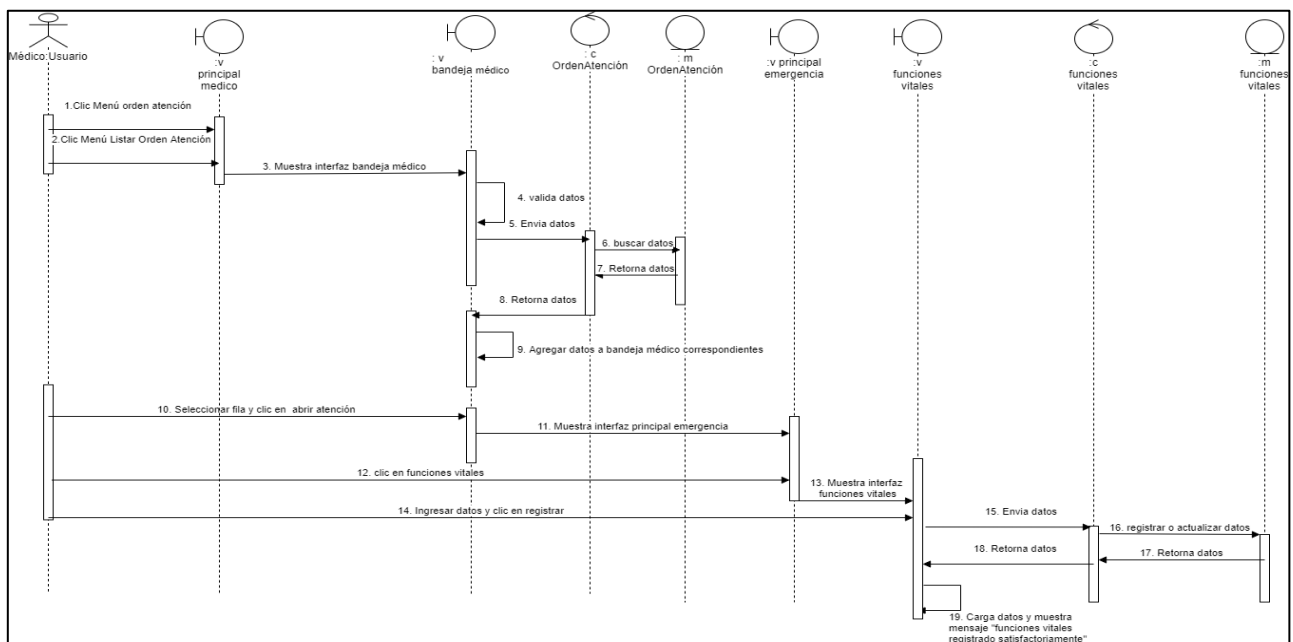
Fuente: Elaboración propia

Figura 40 Diagrama de Secuencia Generar Reporte Triage



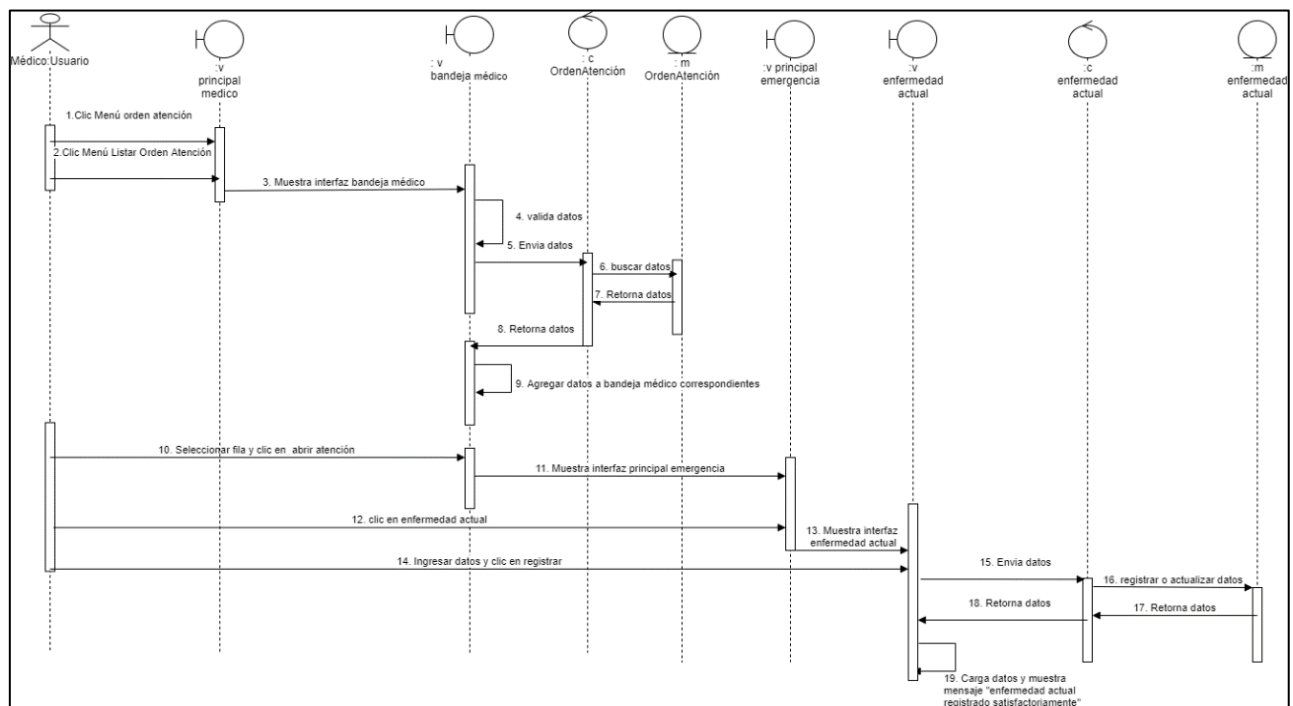
Fuente: Elaboración propia

Figura 42 Diagrama de Secuencia Gestionar Enfermedad Actual



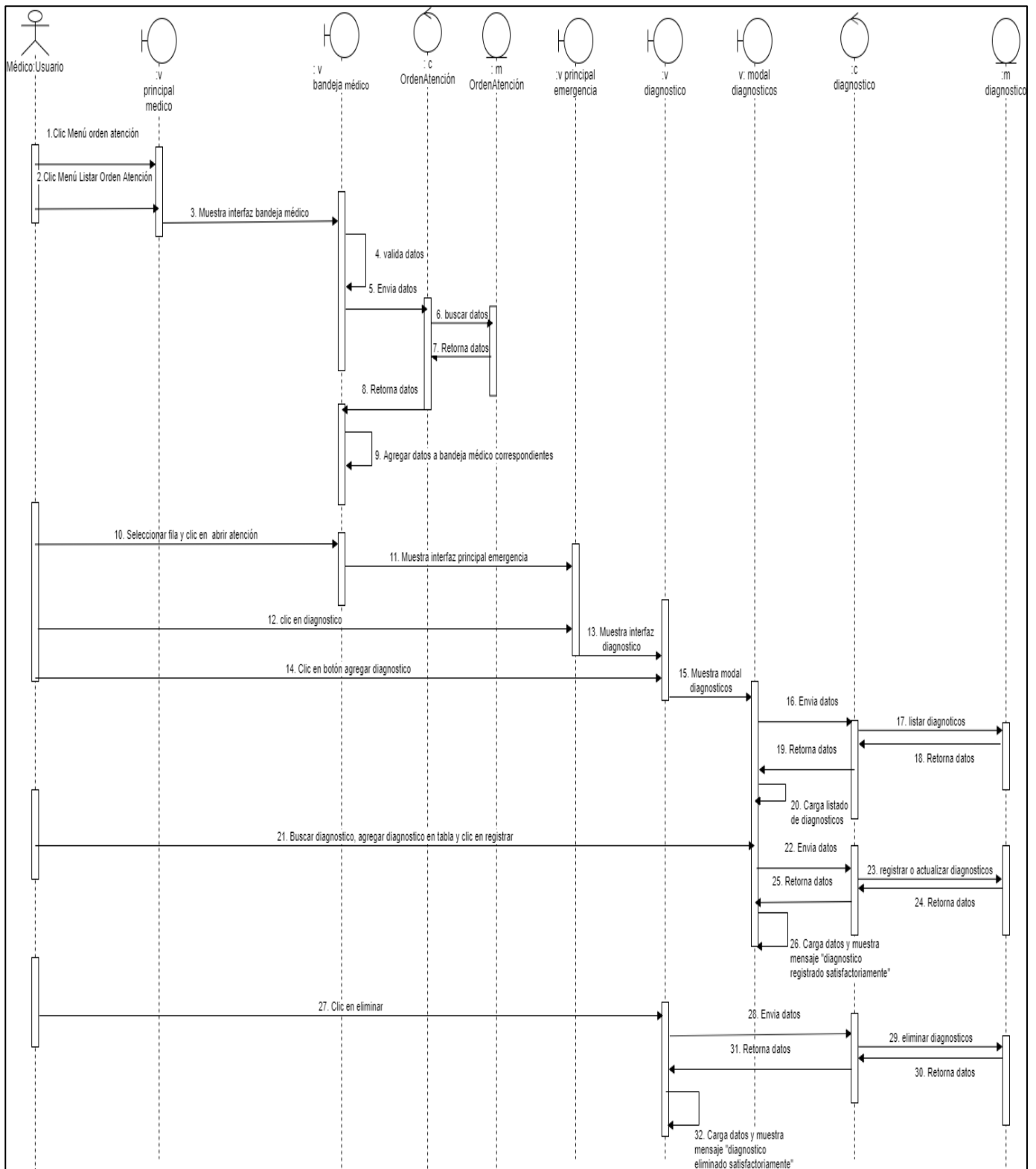
Fuente: Elaboración propia

Figura 41 Diagrama de Secuencia Gestionar Funciones Vitales



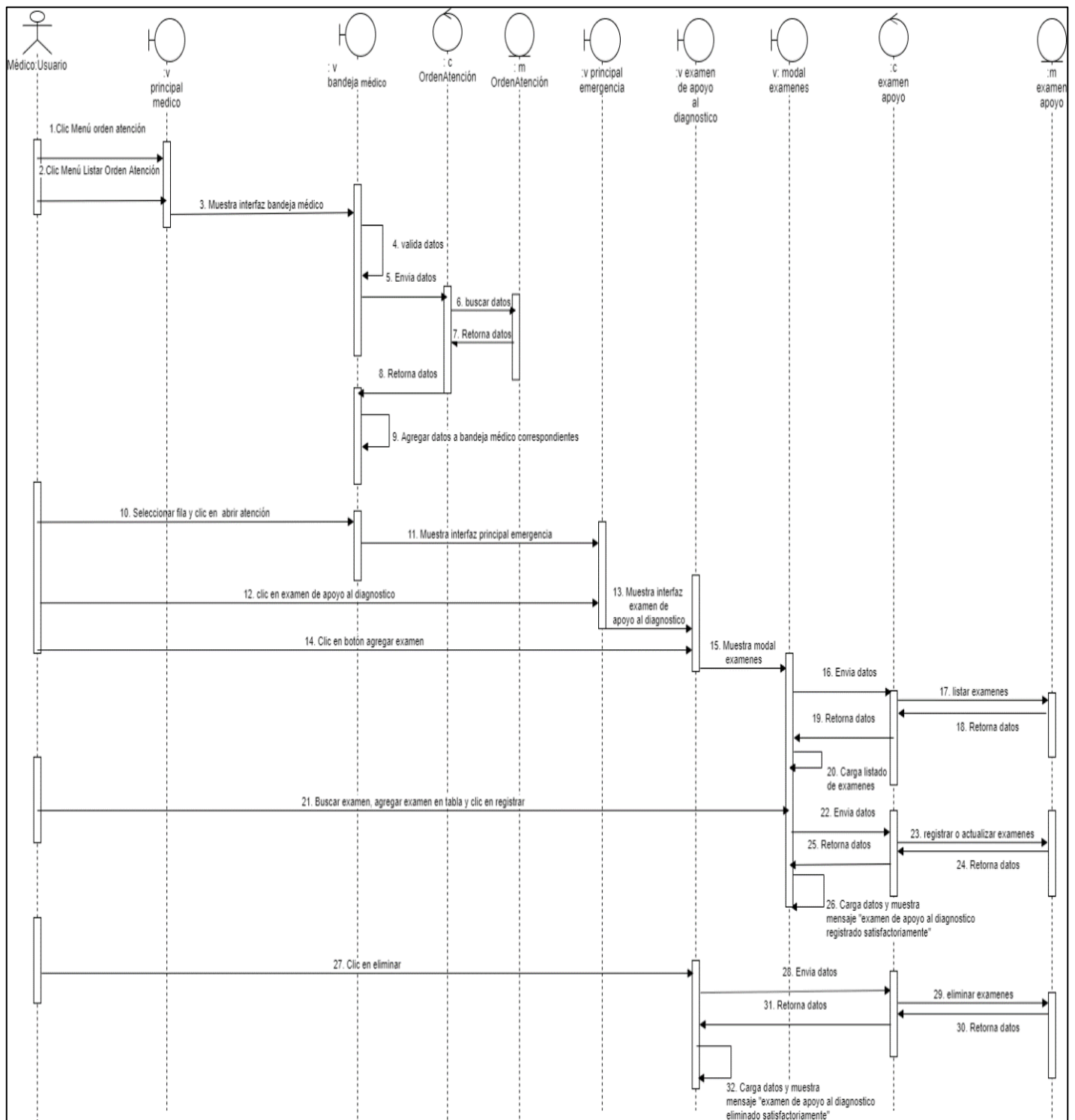
Fuente: Elaboración propia

Figura 43 Diagrama de Secuencia Gestionar Diagnóstico



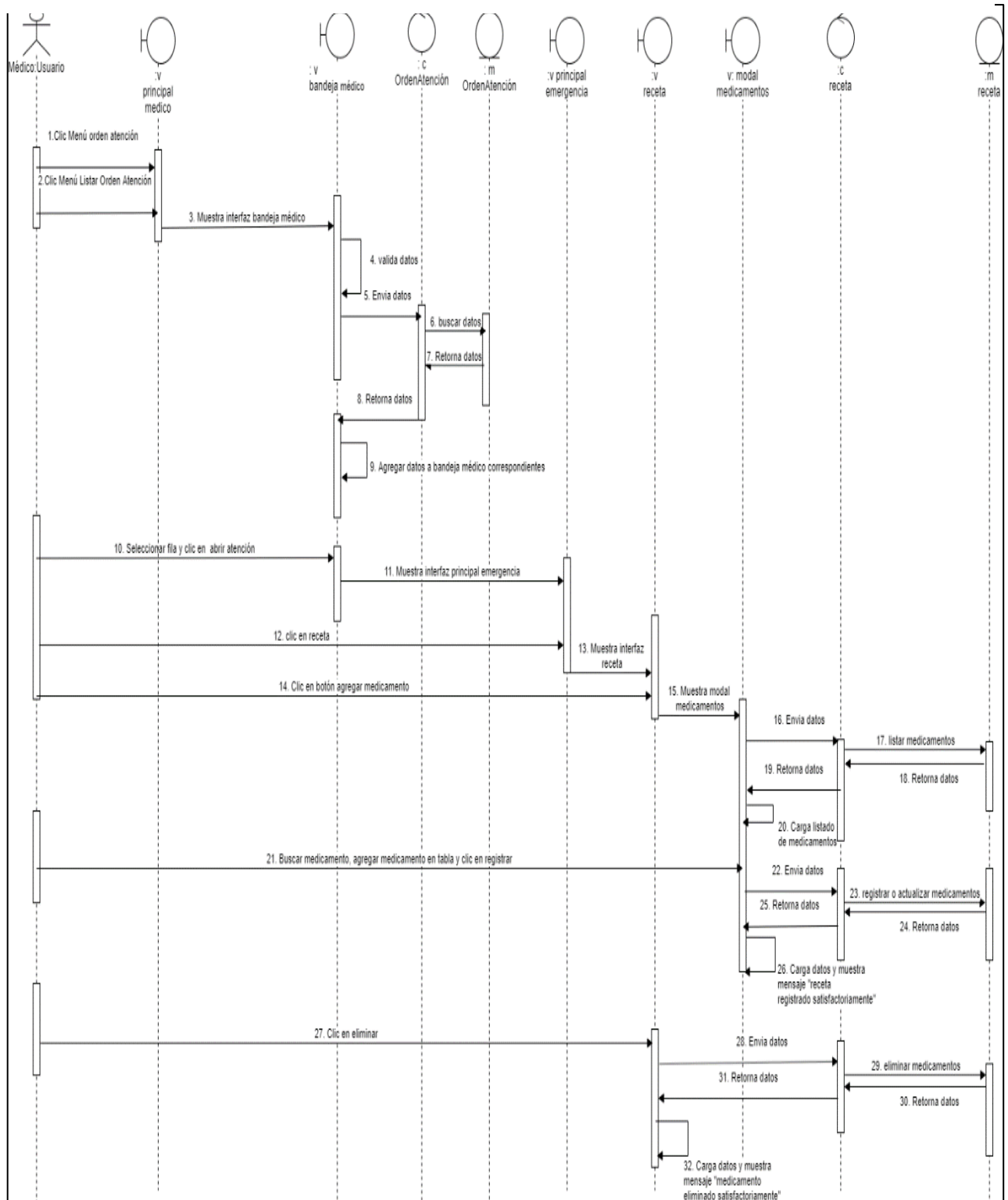
Fuente: Elaboración propia

Figura 44 Diagrama de Secuencia Gestionar Exámenes de Apoyo al Diagnostico



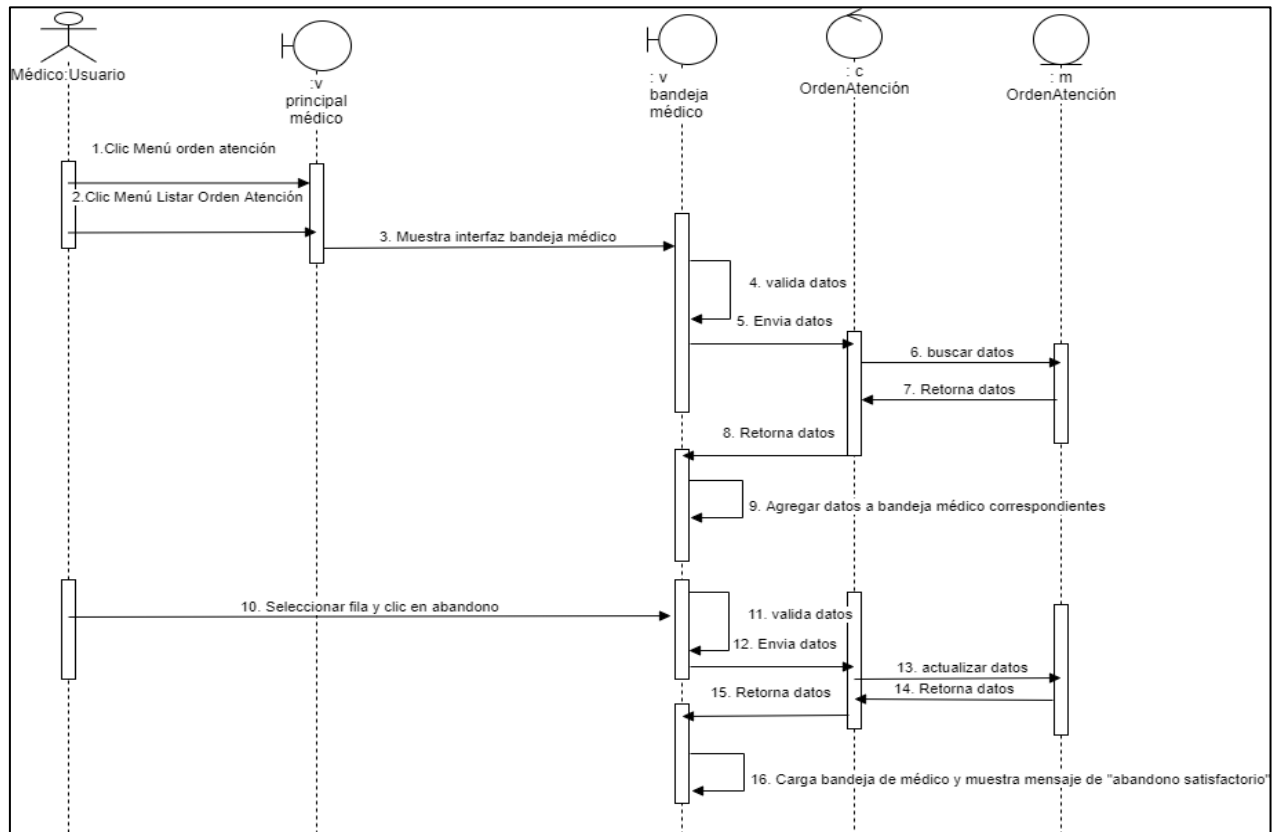
Fuente: Elaboración propia

Figura 45 Diagrama de Secuencia Gestionar Receta



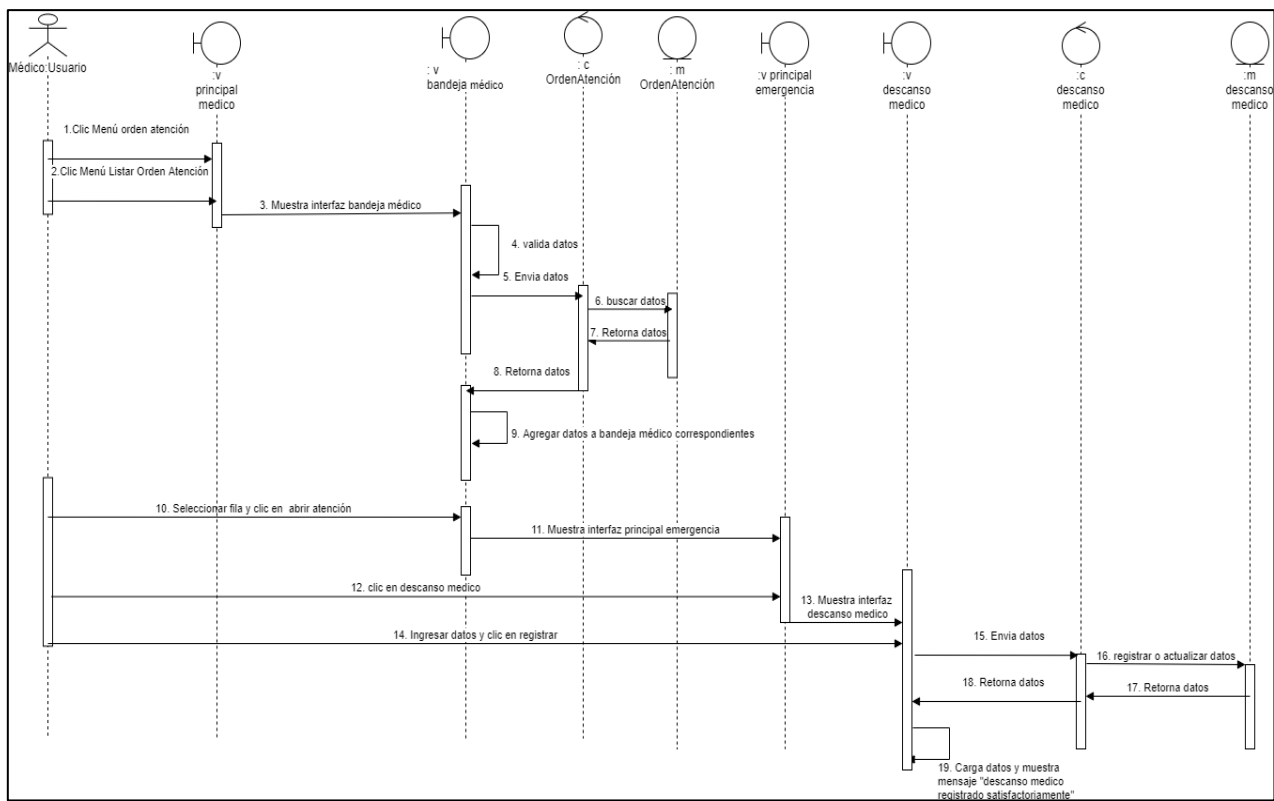
Fuente: Elaboración propia

Figura 47 Diagrama de Secuencia Gestionar Descanso Médico



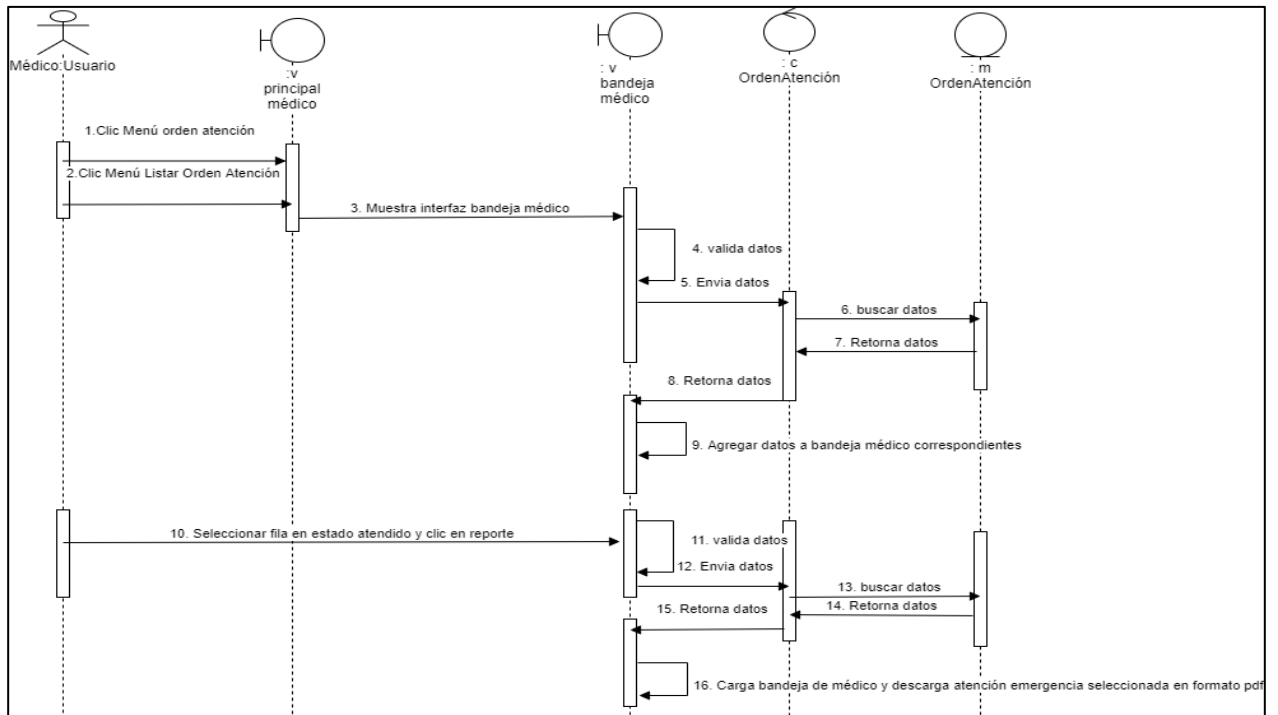
Fuente: Elaboración propia

Figura 46 Diagrama de Secuencia Abandono - Médico



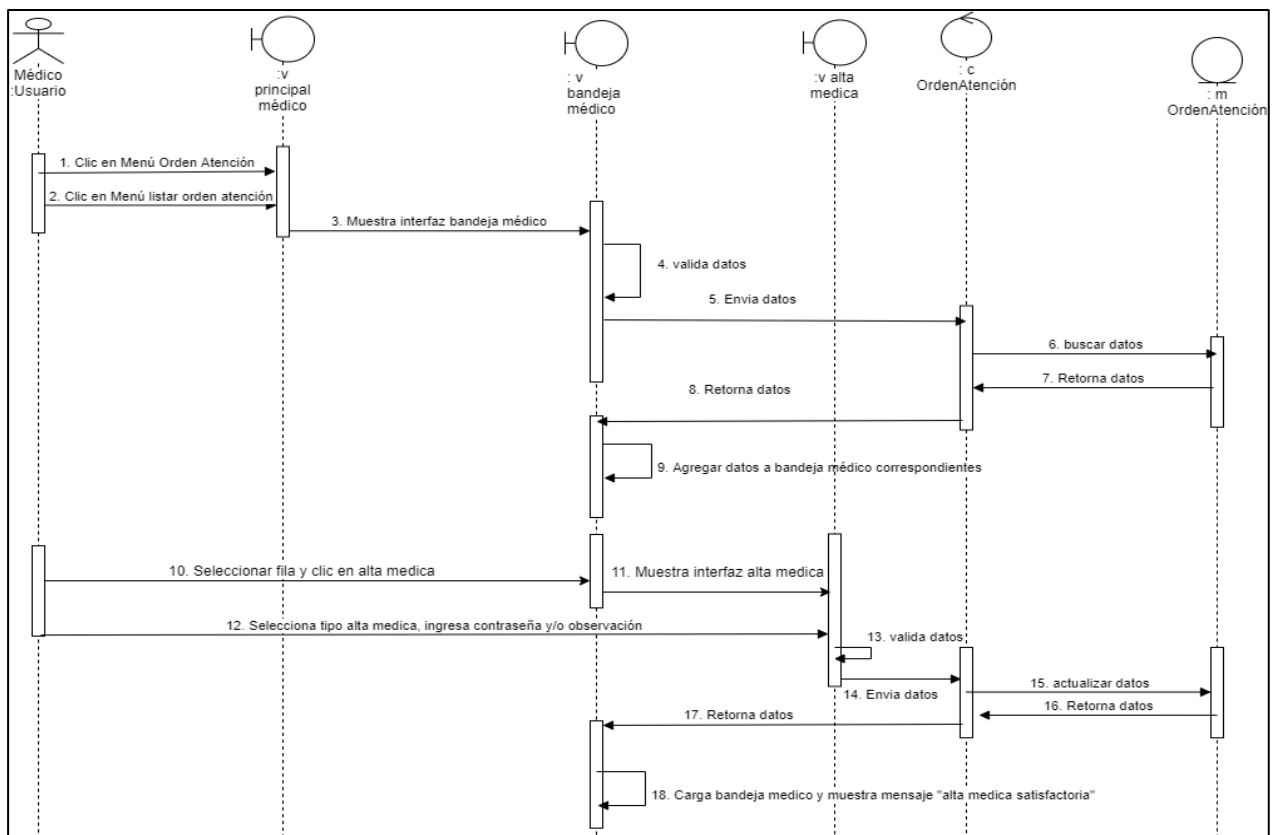
Fuente: Elaboración propia

Figura 49 Diagrama de Secuencia Alta Medica



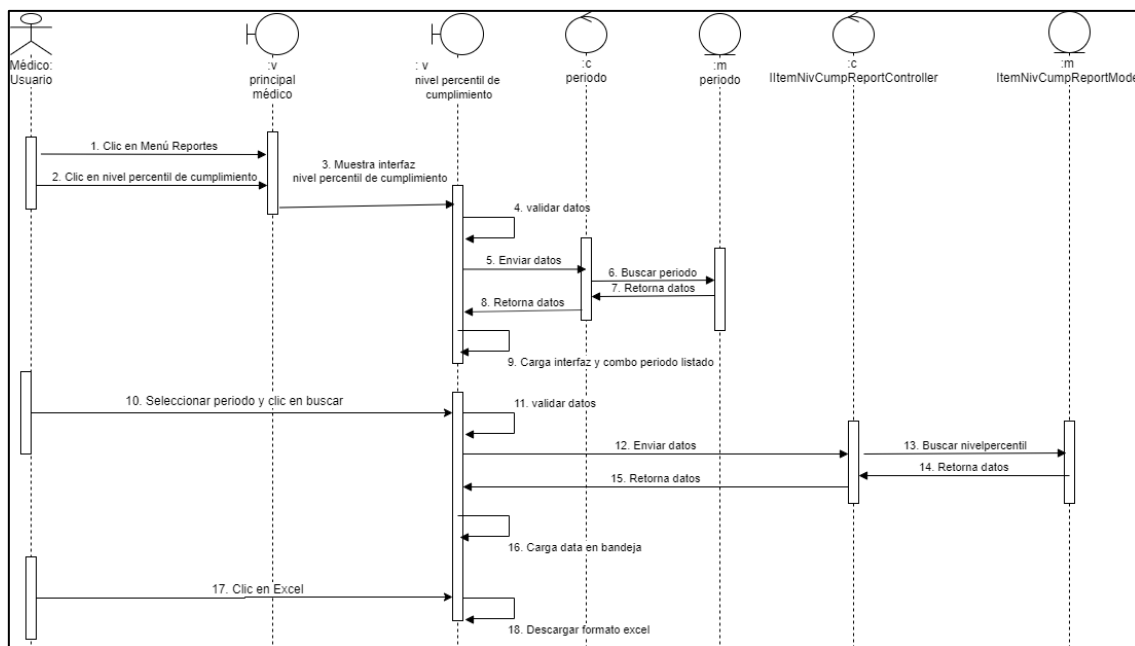
Fuente: Elaboración propia

Figura 48 Diagrama de Secuencia Generar Reporte Atención



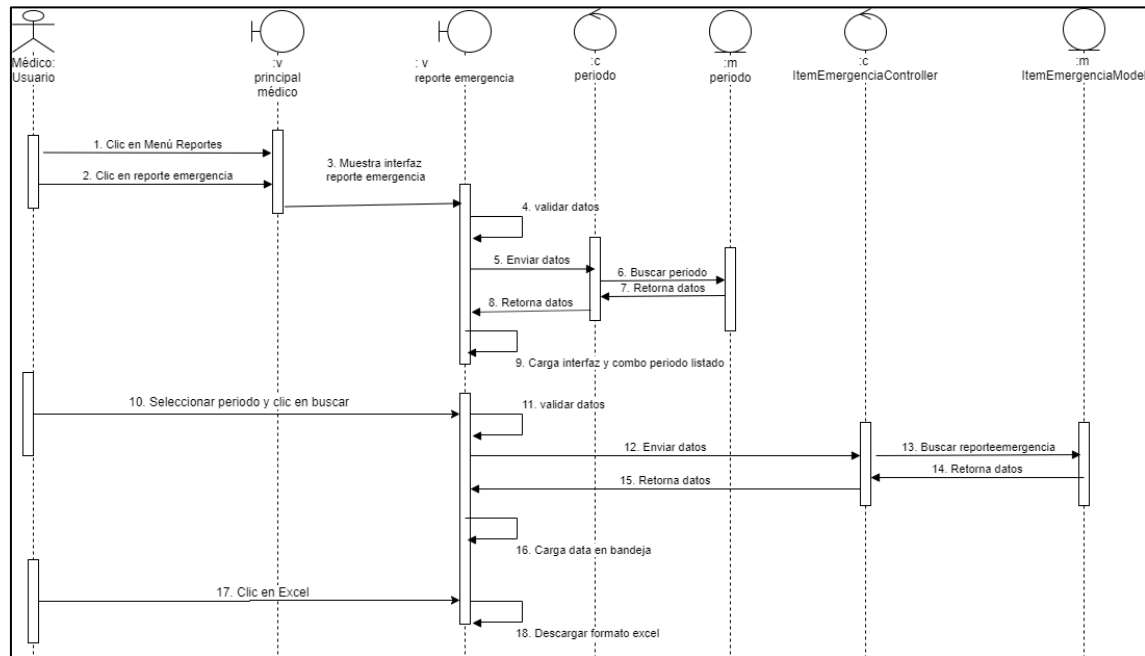
Fuente: Elaboración propia

Figura 50 Generar Reporte Nivel Percentil de Cumplimiento



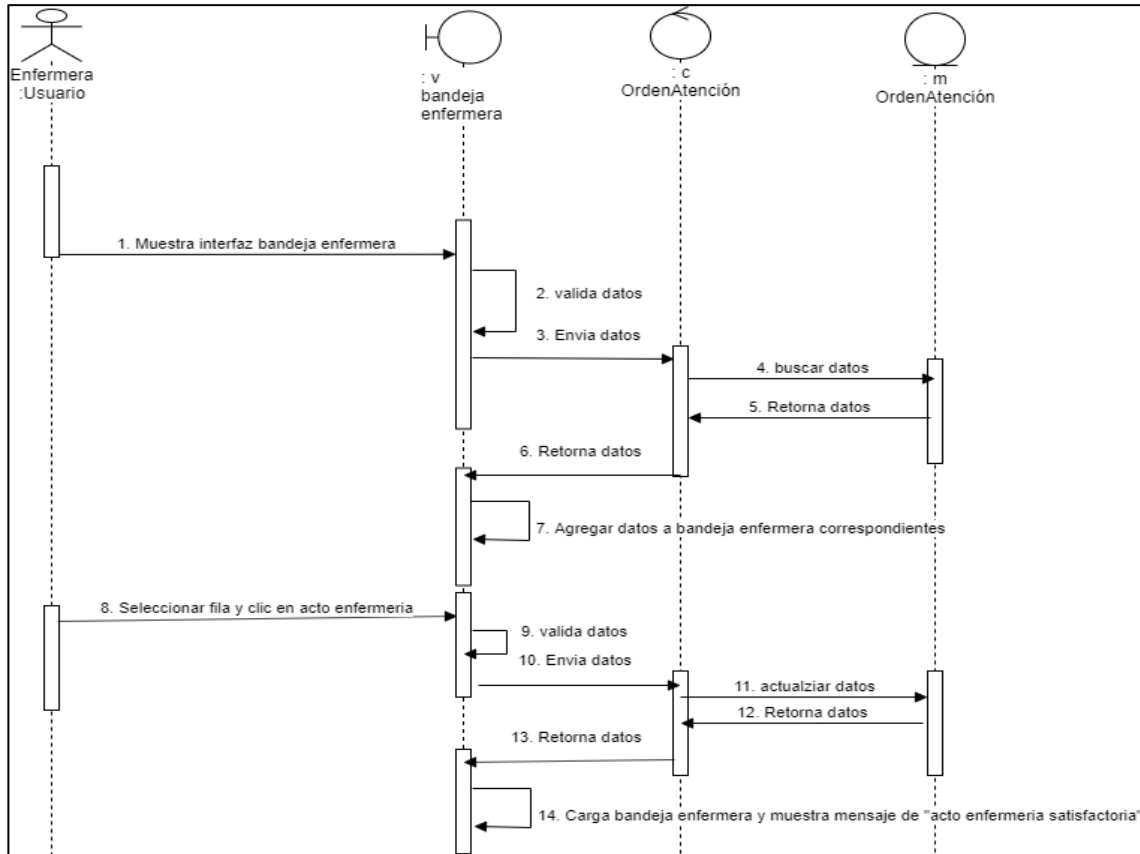
Fuente: Elaboración propia

Figura 51 Diagrama de Secuencia Generar Reporte Emergencia



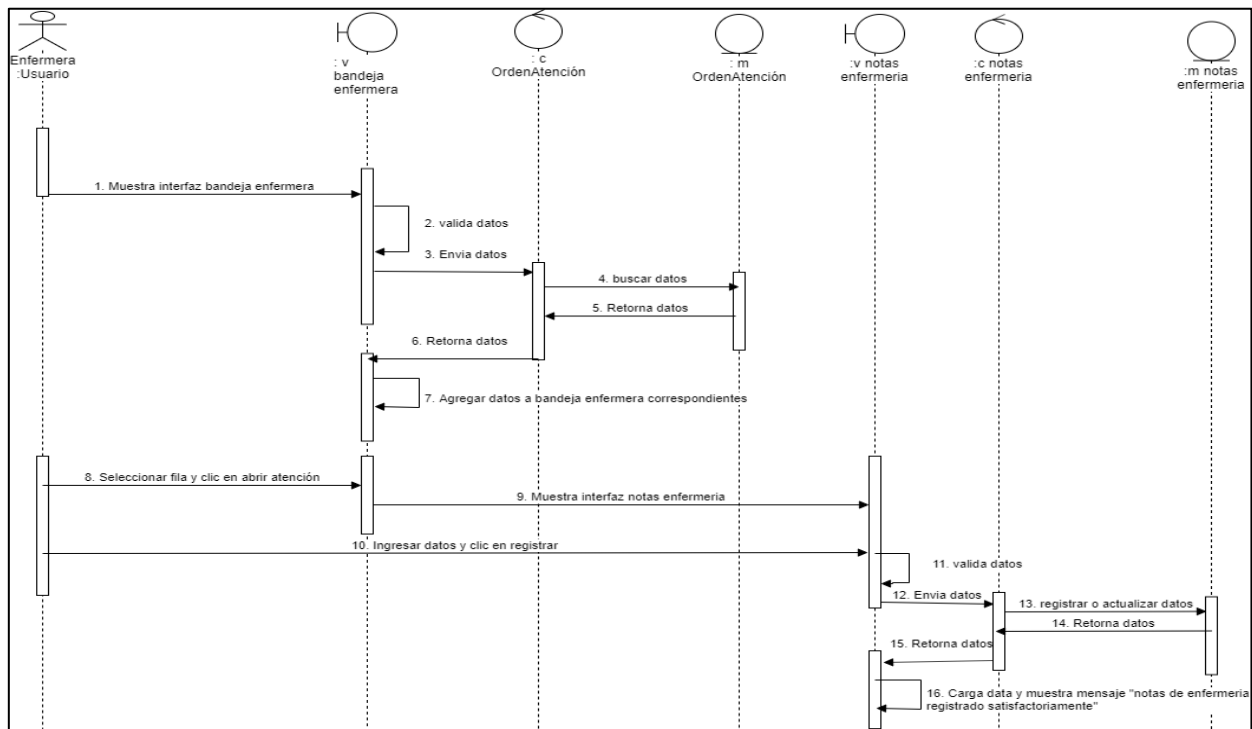
Fuente: Elaboración propia

Figura 53 Diagrama de Secuencia Gestionar Nota Enfermería



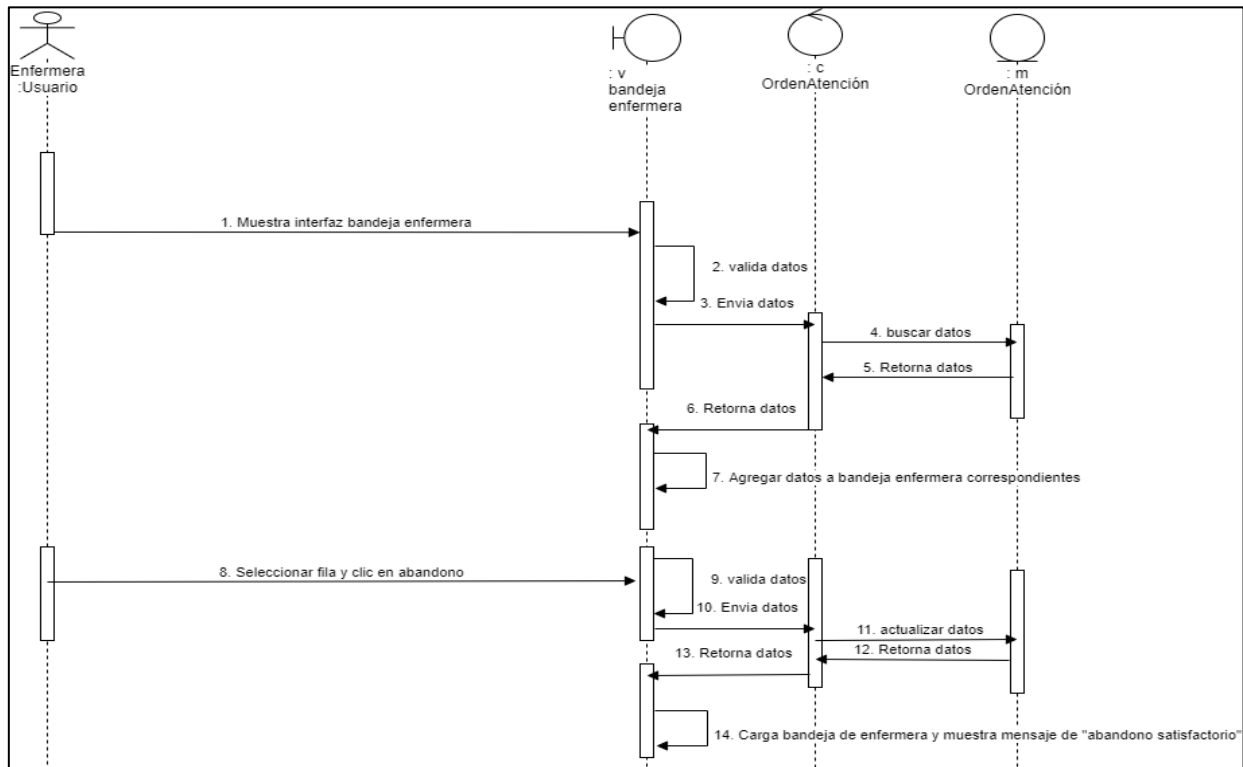
Fuente: Elaboración propia

Figura 52 Diagrama de Secuencia Acto Enfermería



Fuente: Elaboración propia

Figura 54 Secuencia Abandono - Enfermería



Fuente: Elaboración propia

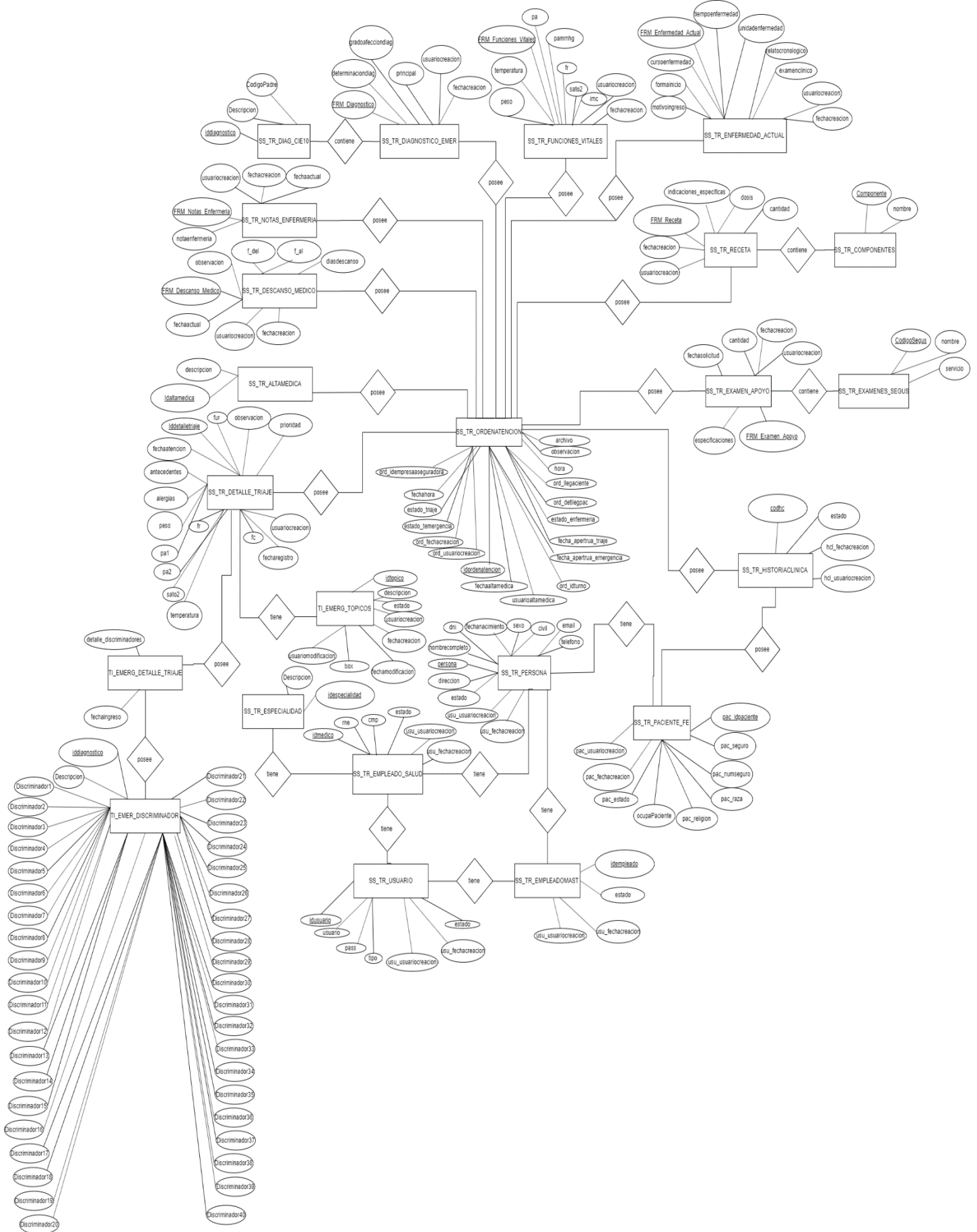
Validación de casos de uso y diagramas de interacción de usuario

Cada usuario valida los casos de uso ya establecidos para asegurarse con la especificación realizada en cada caso de uso, que están dentro del rol que el cumple en el sistema web.

Fase 2: Diseño conceptual

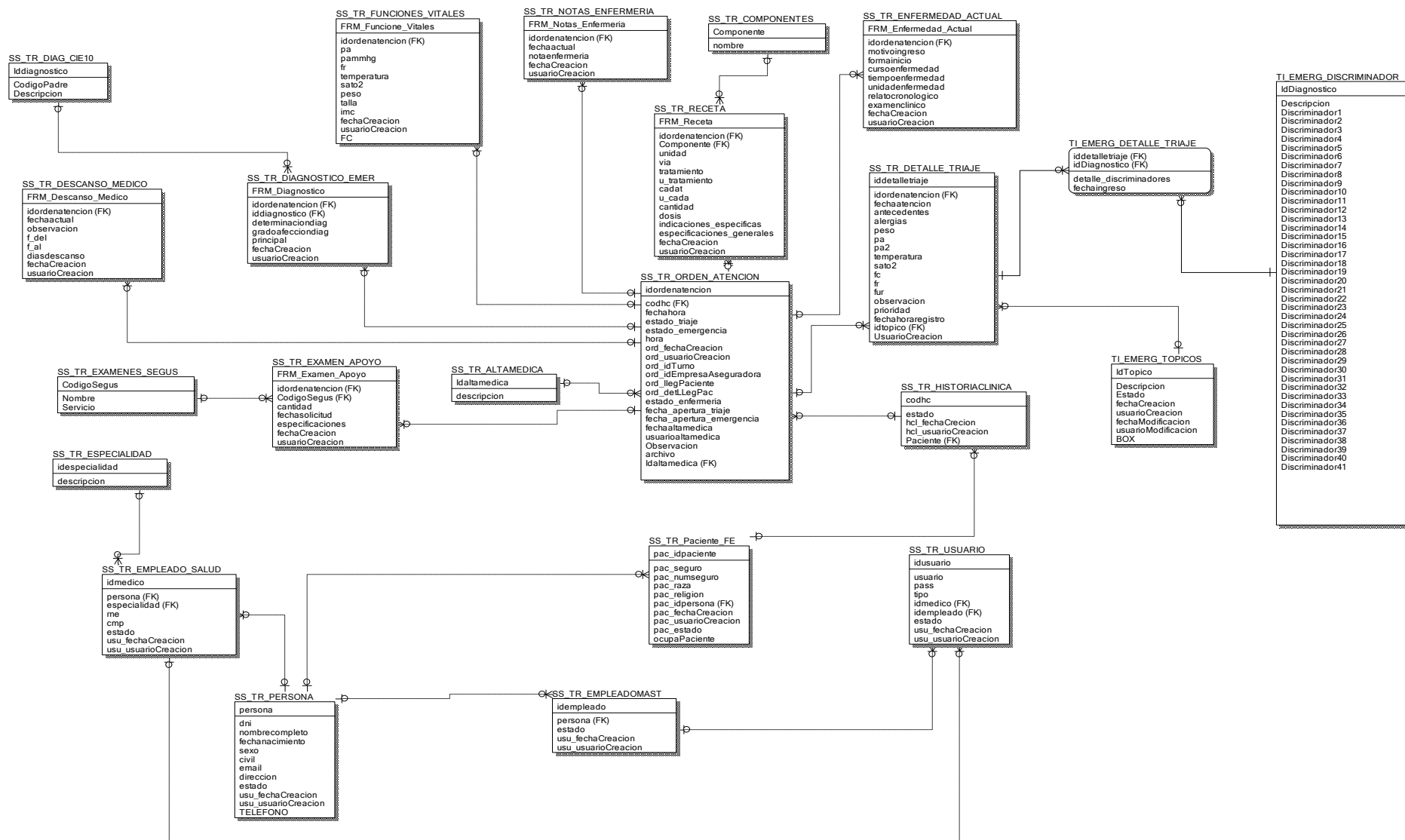
Se tomó en cuenta los requerimientos funcionales de la fase anterior para el desarrollo de la base de datos, obteniendo el modelo de entidad relación, modelo lógico y físico de la base de datos. En la figura 55 se puede ver el diagrama de entidad relación, en donde se ve la interacción de las entidades y atributos. En la figura 56 se puede ver el modelo lógico de la base de datos en donde se aprecia las relaciones entre cada tabla y en la figura 57 se puede ver el modelo físico de la base de datos en donde se aprecia cada campo con su respectivo tipo de dato.

Figura 55 Modelo Entidad Relación



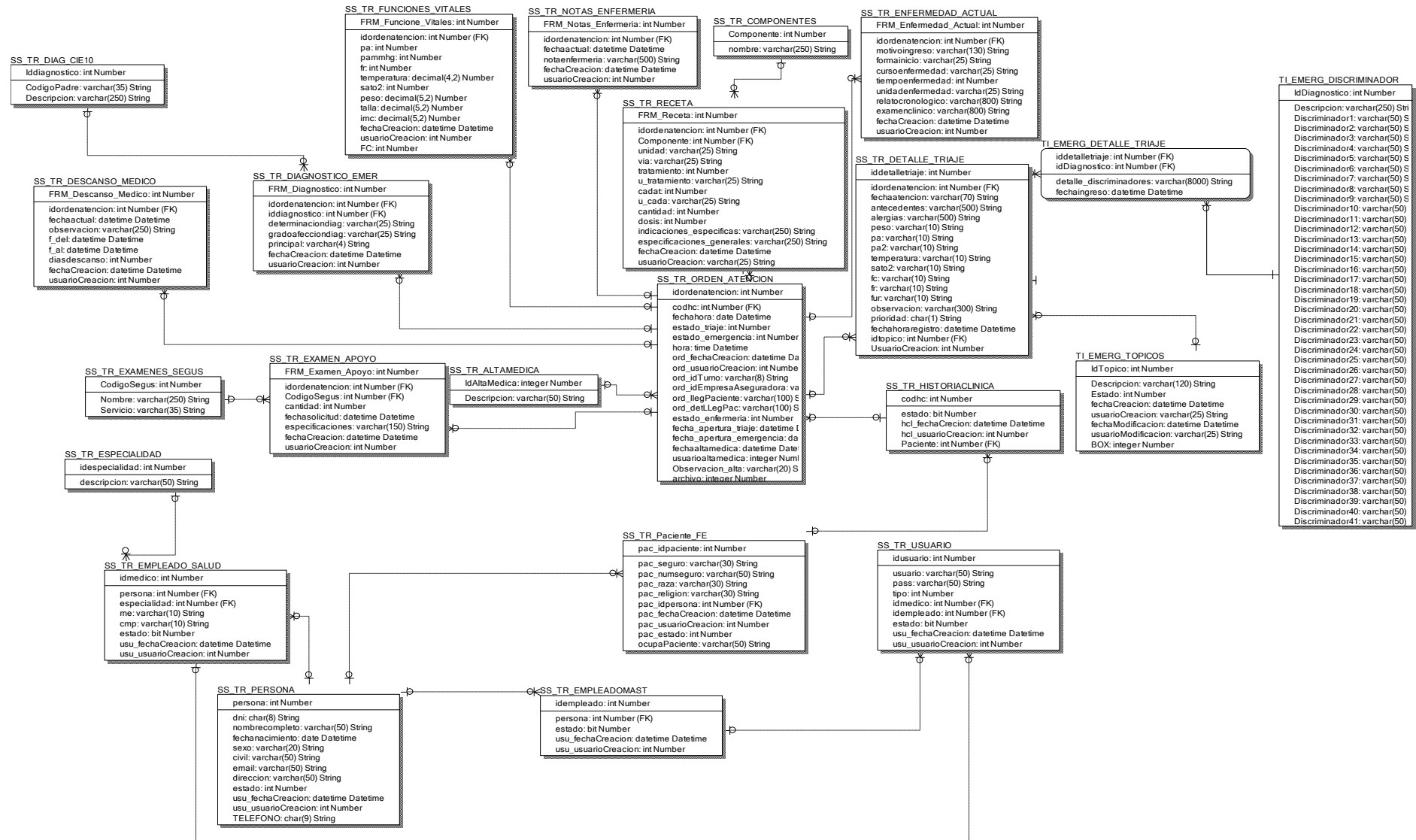
Fuente: Elaboración propia

Figura 56 Diagrama lógico de base de datos



Fuente: Elaboración propia

Figura 57 Diagrama físico de base de datos



Fuente: Elaboración propia

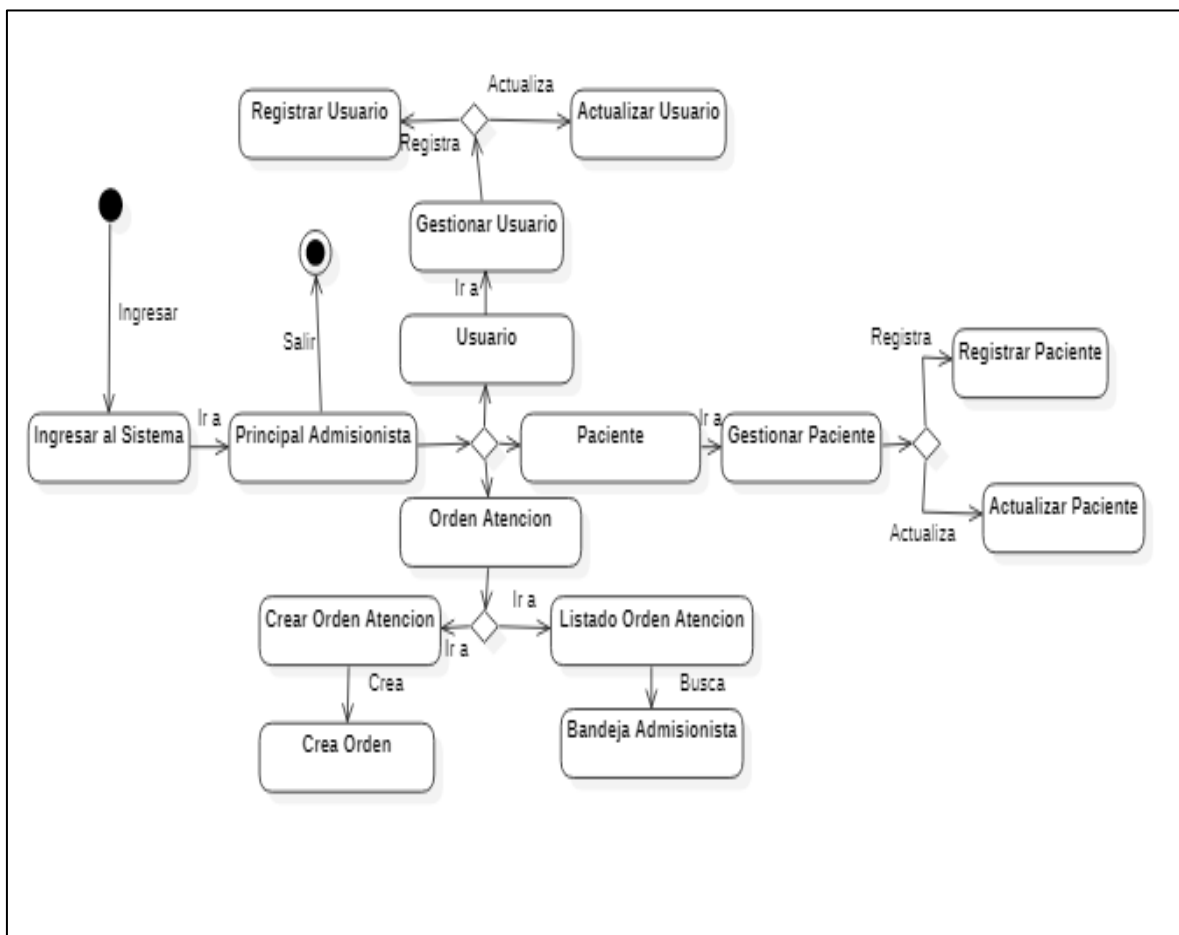
Fase 3: Diseño Navegacional.

En esta etapa se define la estructura de navegación mediante la realización de modelos navegacionales que representan diferentes vistas del modelo conceptual.

En Diseño Navegacional se reorganiza la información para adaptarla a las necesidades de los usuarios del sistema.

En OOHDM, la navegación es considerada un paso crítico en el diseño aplicaciones. Un modelo navegacional es construido como una vista sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual.

Figura 58 Diagrama navegacional de admisionista

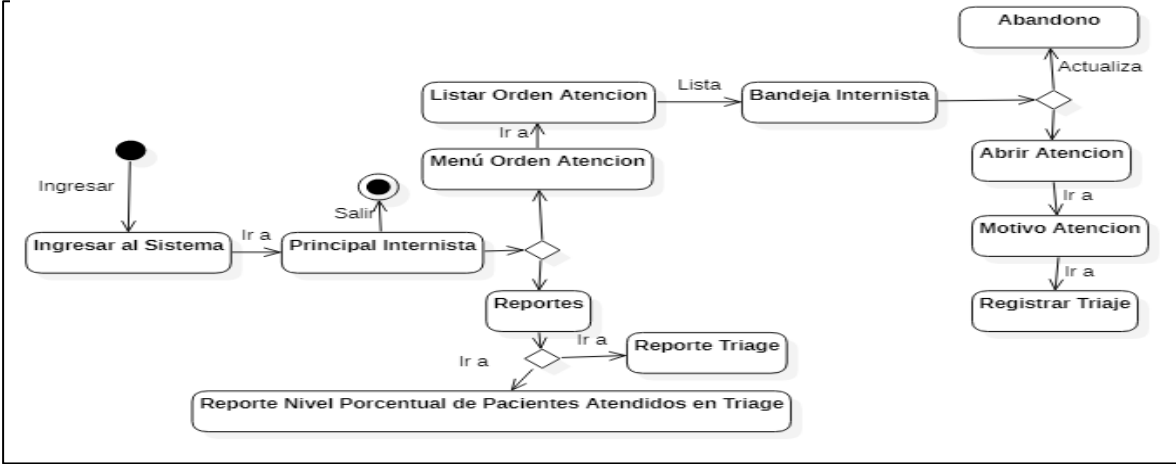


Fuente: Elaboración propia

En la figura 58 se ve el diseño navegacional del admisionista, en donde inicia en la ventana de ingresar al sistema, luego visualiza su pantalla principal en donde muestra los módulos a

los cuales tiene acceso los cuales son: Usuario, Paciente, Orden Atención. De acuerdo a su selección podrá visualizar las ventanas de gestionar usuario y gestionar paciente en donde puede registrar y actualizar, crear orden atención aquí puede crear ordenes de atención para el paciente y listar orden atención en su bandeja admisionista.

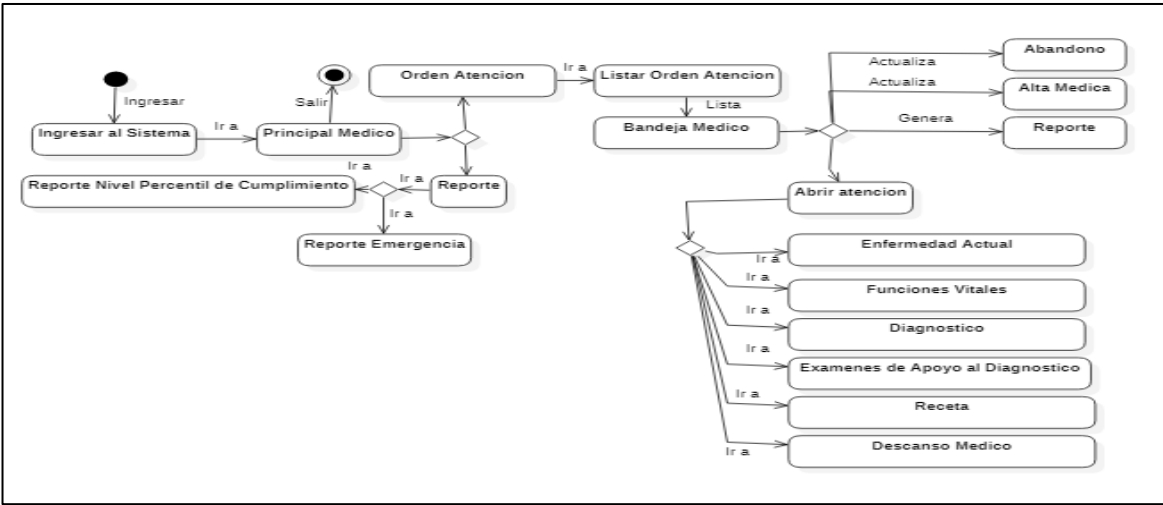
Figura 59 Diagrama navegacional de internista



Fuente: Elaboración propia

En la figura 59 se ve el diseño navegacional del internista, en donde inicia en la ventana de ingresar al sistema luego visualiza su pantalla principal en donde muestra los módulos a los cuales tiene acceso los cuales son: Orden atención y Reportes. De acuerdo a su selección podrá visualizar las ventanas de bandeja internista en donde puede dar por abandono una atención o puede aperturar una atención para luego seleccionar el motivo de atención del paciente en la ventana de motivo atención, finalizando con el registro de triage en la venta de registro triage, además podrá visualizar los reportes de nivel porcentual de pacientes atendidos en triage y reporte de triage.

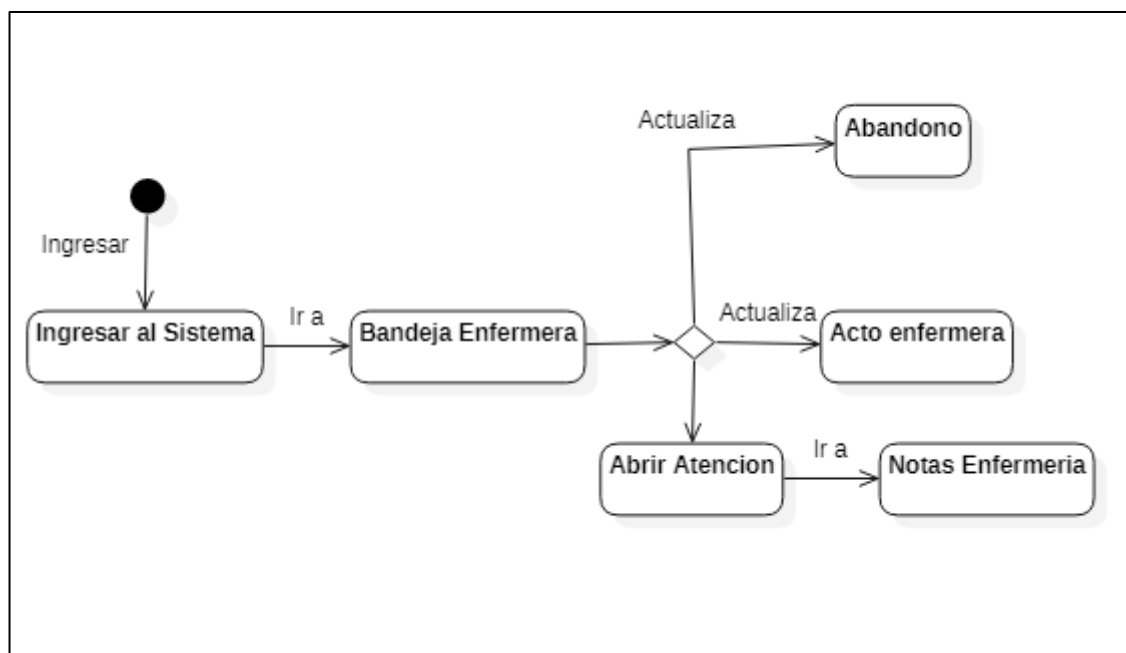
Figura 60 Diagrama navegacional de medico



Fuente: Elaboración propia

En la figura 60 se ve el diseño navegacional del médico en donde inicia en la ventana de ingresar al sistema luego visualiza su pantalla principal en donde muestra los módulos a los cuales tiene acceso los cuales son: Orden atención y Reportes. De acuerdo a su selección podrá visualizar las ventanas de bandeja médico en donde puede dar por abandono una atención, dar alta médica al finalizar una atención o puede aperturar una atención para luego realizar el registro de emergencia en los distintos formatos tales como: enfermedad actual, funciones vitales, diagnóstico, exámenes de apoyo al diagnóstico, receta y descanso médico, además podrá visualizar los reportes de nivel percentil de cumplimiento y reporte de emergencia.

Figura 61 Diagrama navegacional de enfermera



Fuente: Elaboración propia

En la figura 60 se ve el diseño navegacional de la enfermera en donde inicia en la ventana de ingresar al sistema luego visualiza su pantalla principal en donde muestra la bandeja de enfermera ya cargada. En la bandeja de enfermera puede dar por abandono una atención, dar un acto enfermería al finalizar una atención o puede aperturar una atención para luego realizar el registro de sus notas de enfermería.

Fase 4: Diseño de Interfaces Abstractas.

En esta fase se especifica las interfaces abstractas del sistema web, indica la maquetación en el cual serán mostrado las interfaces.

Figura 62 Diseño de Interfaz Ingresar al

A Web Page

https://

SANNA
CLÍNICA SAN BORJA

USUARIO

CONTRASEÑA

ComboBox ▼

LOGIN

Fuente: Elaboración propia

En la Figura 62 observamos el prototipo de ingresar al sistema en donde los usuarios (admisionista, internista, médico y enfermera) ingresarán a sus distintos módulos de atención especificando su usuario, contraseña y tipo de usuario.

Figura 63 Diseño de Interfaz Gestionar Usuario

A Web Page

https://

USUARIO >
PACIENTE >
ORDEN ATENCION >

GESTION USUARIO

TIPO USUARIO

DNI 🔍

NOMBRE COMPLETO

FECHA NACIMIENTO / / SEXO

ESTADO CIVIL CORREO ELECTRONICO

DIRECCION ESTADO

ESPECIALIDAD N° COLEGIATURA RNE

TELEFONO CONTRASEÑA

REGISTRAR ACTUALIZAR

Fuente: Elaboración propia

En la Figura 63 observamos la interfaz de gestionar usuarios, en el cual podremos registrar los usuarios admisionistas, internistas, médicos y enfermeros ingresando los datos correspondientes dependiendo del tipo de empleado, además nos permitirá la búsqueda del DNI para luego actualizar los datos necesarios.

Figura 64 Diseño Interfaz Gestionar Pacientes

Fuente: Elaboración propia

En la Figura 64 observamos la interfaz de gestionar pacientes, en el cual podremos registrar los pacientes ingresando los datos correspondientes, nos permitirá la búsqueda del DNI para luego actualizar los datos necesarios.

Figura 65 Diseño Interfaz Crear Orden Atención

Fuente: Elaboración propia

En la Figura 65 observamos la interfaz de crear de orden atención, en donde podremos realizar la búsqueda del DNI y se ingresa el ingreso del paciente al área de emergencia y realizamos la creación de la orden atención.

Figura 66 Diseño Interfaz Listar Ordenes de Atención en Bandeja Admisionista

The interface is titled 'BANDEJA ADMISIONISTA'. It includes a left sidebar with a menu containing 'USUARIO', 'PACIENTE', and 'ORDEN ATENCION'. The main content area has two date filter fields labeled 'FECHA INICIO' and 'FECHA FIN', each with a calendar icon. To the right of these filters is a magnifying glass icon for search. Below the filters is a table with the following columns: OrdenAtencion, CodHC, Nombre Paciente, Sexo, Edad, Fecha, Hora, and Estado.

Fuente: Elaboración propia

En la Figura 66 observamos la interfaz del listado de órdenes atención en la bandeja admisionista para poder visualizar la cola de pacientes en espera de su atención y en donde podremos listar las atenciones en estado pendiente y en atención.

Figura 67 Diseño Interfaz Listar Ordenes de Atención en Bandeja Internista

The interface is titled 'BANDEJA INTERNISTA'. It includes a left sidebar with a menu containing 'ORDEN ATENCION' and 'REPORTES'. The main content area has two date filter fields labeled 'FECHA INICIO' and 'FECHA FIN', each with a calendar icon. To the right of these filters is a magnifying glass icon for search and a button labeled 'ABANDONO'. Below the filters is a table with the following columns: OrdenAtencion, CodHC, Nombre Paciente, Sexo, Edad, Fecha, Hora, Estado, and Abrir Atencion.

Fuente: Elaboración propia

En la Figura 67 observamos la interfaz del listado de órdenes atención en espera de triaje en donde podremos visualizar el listado en la bandeja internista, además lista las atenciones en estado pendiente y en atención, también podremos seleccionar una fila de la bandeja internista para actualizar su estado a abandono en los casos que el paciente no se presente a la atención.

Figura 68 Diseño Interfaz Listar Ordenes de Atención en Bandeja Medico

The screenshot shows a web browser window titled 'A Web Page' with the URL 'https://'. The page is titled 'BANDEJA MEDICO'. On the left, there is a navigation menu with 'ORDEN ATENCION' and 'REPORTES'. The main content area has filters for 'FECHA INICIO' and 'FECHA FIN' with calendar icons, a search icon, and buttons for 'ABANDONO', 'ALTA MEDICA', and 'REPORTE'. Below these is a table with the following columns: OrdenAtencion, CodHC, Nombre Paciente, Sexo, Edad, Fecha, Hora, Estado, and Abrir Atencion.

Fuente: Elaboración propia

En la Figura 68 observamos la interfaz del listado de órdenes atención en espera de la atención médica, podremos seleccionar una fila de la bandeja médico para realizar una alta médica para terminar su atención y cambiar el estado de la orden a atendido, también podremos seleccionar una fila de la bandeja médico para actualizar su estado a abandono en los casos que el paciente no se presente a la atención. Podremos visualizar en el listado todos los estados menos el abandonado (pendiente, atendido, en atención).

Figura 69 Diseño Interfaz Listar Ordenes de Atención en Bandeja Enfermera

The screenshot shows a web browser window titled 'A Web Page' with the URL 'https://'. The page is titled 'BANDEJA ENFERMERA'. On the left, there is a navigation menu with 'ORDEN ATENCION' and 'REPORTES'. The main content area has filters for 'FECHA INICIO' and 'FECHA FIN' with calendar icons, a search icon, and buttons for 'ABANDONO' and 'ACTO ENFERMERA'. Below these is a table with the following columns: OrdenAtencion, CodHC, Nombre Paciente, Sexo, Edad, Fecha, Hora, Estado, and Abrir Atencion.

Fuente: Elaboración propia

En la Figura 69 observamos la interfaz del listado de órdenes atención en espera de la atención de enfermería, podremos seleccionar una fila de la bandeja enfermera para realizar su acto de enfermería para actualizar el estado de la orden de en atención a atendido, además podremos seleccionar una fila de la bandeja enfermera para actualizar su estado a abandono en los casos que el paciente no se presente a la atención.

Figura 70 Diseño de Interfaz Motivos Atención

The image shows a web browser window titled 'A Web Page' with a URL bar containing 'https://'. The main content area displays a grid of 32 rectangular buttons, each labeled 'Motivo#'. The buttons are arranged in 4 columns and 8 rows.

Fuente: Elaboración propia

En la Figura 70 observamos la interfaz de motivo atención (diagnostico o dolencia del paciente) en donde podremos seleccionar un motivo de atención de acuerdo a lo que comenta el paciente y así poder empezar con la atención de triaje.

Figura 71 Diseño de Interfaz Registrar Triage

The image shows a web browser window titled 'A Web Page' with a URL bar containing 'https://'. The main content area displays a form for 'Registrar Triage'. The form includes several sections:

- Prioridad:** A dropdown menu labeled 'ComboBox'.
- Topico:** A dropdown menu labeled 'ComboBox'.
- Paciente:** Fields for 'FechaIngreso', 'HoraIngreso', 'Edad', 'Sexo', and 'Motivo Ingreso'.
- Signos Vitales:** Fields for 'Peso', 'Kg', 'PA', 'T°', 'SATO2', 'FC', 'X Min', and 'FR'.
- Antecedentes:** A large text area labeled 'ANTECEDENTES'.
- Observación:** A large text area labeled 'OBSERVACION'.
- Buttons:** A 'REGISTRAR TRIAJE' button and a 'FUR' button.

Fuente: Elaboración propia

En la Figura 71 observamos la interfaz de registro de triaje en donde podremos seleccionar los discriminadores necesarios para que el sistema valide la prioridad, además registramos los signos vitales, alergias, antecedentes y el tópico disponible a derivar al paciente.

Figura 72 Diseño de Interfaz Reporte Nivel Porcentual Pacientes Atendidos en Triage

ORDEN ATENCION >
REPORTES >

REPORTE NIVEL PORCENTUAL PACIENTES ATENDIDOS EN TRIAGE

Seleccione Periodo:

Item	Fecha	Internista	Turno	CTP	CPC	NPCA

Fuente: Elaboración propia

En la Figura 72 observamos la interfaz de reporte nivel porcentual de pacientes atendidos en triaje en donde se podremos seleccionar el periodo del mes a buscar y luego nos permite exportar esa data a un formato Excel.

Figura 73 Diseño de Interfaz Reporte Triage

ORDEN ATENCION >
REPORTES >

REPORTE TRIAGE

Seleccione Periodo:

HC	Seguro	Paciente	DNI	Sexo	Edad	Triage	Motivo	Discriminador	Internista	Prioridad	Topico	Apertura Triage	Registro Triage

Fuente: Elaboración propia

En la Figura 73 observamos la interfaz de reporte triage en donde se podremos seleccionar el periodo del mes a buscar y luego nos permite exportar esa data a un formato Excel.

Figura 74 Diseño de Interfaz Gestionar Enfermedad Actual

Fuente: Elaboración propia

En la Figura 74 observamos la interfaz del formato de enfermedad actual en donde el podremos registrar el motivo ingreso y características de la enfermedad o dolencia del paciente.

Figura 75 Diseño Interfaz Gestionar Funciones Vitales

Fuente: Elaboración propia

En la Figura 75 observamos la interfaz de las funciones vitales en donde podremos registrar el estado vital del paciente.

Figura 76 Diseño de Interfaz Gestionar Diagnostico

A Web Page

https://

ENFERMEDAD ACTUAL

FUNCIONES VITALES

DIAGNOSTICO

EXÁMENES DE APOYO AL DIAGNOSTICO

RECTA

DESCANSO MEDICO

IdDiagnostico Diagnostico DeterminacionDiagnostico GradoAfeccion Principal Eliminar

REGISTRAR

Agregar Diagnosticos

Fuente: Elaboración propia

En la Figura 76 observamos la interfaz de diagnóstico en donde podremos registrar los diagnósticos relacionados al motivo de atención.

Figura 77 Diseño Interfaz Gestionar Examen de Apoyo al Diagnostico

A Web Page

https://

ENFERMEDAD ACTUAL

FUNCIONES VITALES

DIAGNOSTICO

EXÁMENES DE APOYO AL DIAGNOSTICO

RECTA

DESCANSO MEDICO

CodigoSegus Examen Cantidad Especificaciones Eliminar

REGISTRAR

Agregar Examen

Fuente: Elaboración propia

En la Figura 77 observamos la interfaz de exámenes de apoyo al diagnóstico en donde podremos registrar los exámenes para la evaluación del paciente en las áreas de laboratorio, imágenes, etc.

Figura 78 Diseño de Interfaz Gestionar Receta

CodigoComponente	Nombre	Cantidad	Dosis	Especificacion
------------------	--------	----------	-------	----------------

Fuente: Elaboración propia

En la Figura 78 observamos la interfaz de receta en podremos registrar los medicamentos necesarios para el paciente.

Figura 79 Diseño de Interfaz Gestionar Descanso

Del / / Al / /

Dias Calculados

Fuente: Elaboración propia

En la Figura 79 observamos la interfaz de descanso médico en donde podremos registrar un tiempo de descanso para el paciente.

Figura 80 Diseño de Interfaz Reporte Nivel Percentil de Cumplimiento

ORDEN ATENCION
REPORTES

REPORTE NIVEL PERCENTIL DE CUMPLIMIENTO

Seleccione Periodo

Item	Fecha	Medico	Area Emergencia	Turno	TPAT	P1	P2	P3	P4	P5	NPC

Fuente: Elaboración propia

En la Figura 80 observamos la interfaz de alta médica en donde podremos realizar el cambio de atención a atendido de la atención de emergencia ingresando la contraseña y tipo de alta médica.

Figura 81 Diseño de Interfaz Alta Médica

ALTA MEDICA

Medico

Usuario

Contraseña

Tipo de alta

Archivo

Observacion

Fuente: Elaboración propia

En la Figura 81 observamos la interfaz de reporte nivel percentil de cumplimiento en donde se podremos seleccionar el periodo del mes a buscar y luego nos permite exportar esa data a un formato Excel.

Figura 82 Diseño de Interfaz Reporte Emergencia

HC	Seguro	Paciente	Emergencia	Medico	Prioridad	Box	Diagnostico	Diagnostico	Diagnostico	Registro Triage	Apertura Emergencia	Fecha Alta	Alta Medica

Fuente: Elaboración propia

En la Figura 82 observamos la interfaz de reporte emergencia en donde se podremos seleccionar el periodo del mes a buscar y luego nos permite exportar esa data a un formato Excel.

Figura 83 Diseño de Interfaz Gestionar Notas de Enfermería

Fuente: Elaboración propia

En la Figura 83 observamos la interfaz de notas de enfermería en donde podremos ingresar información sobre el estado del paciente.

Fase 5: Implementación.

Codificación

En esta etapa se realizó la codificación necesaria para el funcionamiento de los requerimientos funcionales. Se inició con la construcción de cada interfaz, luego se realizó la codificación de validaciones y funcionamiento, finalmente la conexión con la base de datos Sql Server. A continuación, se presentará la codificación más relevante en el desarrollo del sistema web.

Figura 84 Modelo Usuario

```
package pe.edu.sanna.triage.model;

import pe.edu.sanna.triage.bean.PersonaBean;

public class UsuarioModel extends PersonaBean{
    String usuario;
    String pass;
    int estado;
    String tipo;
    String usuarioCreacion_use;

    public String getUsuarioCreacion_use() {
        return usuarioCreacion_use;
    }
    public void setUsuarioCreacion_use(String usuarioCreacion_use) {
        this.usuarioCreacion_use = usuarioCreacion_use;
    }
    public String getUsuario() {
        return usuario;
    }
    public void setUsuario(String usuario) {
        this.usuario = usuario;
    }
    public String getPass() {
        return pass;
    }
    public void setPass(String pass) {
        this.pass = pass;
    }
    public int getEstado() {
        return estado;
    }
    public void setEstado(int estado) {
        this.estado = estado;
    }
    public String getTipo() {
        return tipo;
    }
    public void setTipo(String tipo) {
        this.tipo = tipo;
    }
}
```

Fuente: Elaboración propia

En la figura 84 se puede ver el código del modelo usuario, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase UsuarioModel.

Figura 85 Vista Ingresar al Sistema

```
$.ajax({
  type: "POST",
  url: "${pageContext.request.contextPath}/Usuario/Login/"+user+"-"+pass+"-"+tipo,
  crossDomain: true,
  data:{},
  contentType: "application/json; charset=utf-8",
  dataType: "json",
  async: true,
  success: function (response) {

    if(response.status === true){

      $(location).attr("href", "${pageContext.request.contextPath}/OrdenAtencion/Web");
    }else{

      $.toast({
        heading: 'Aviso!',
        text: 'USUARIO, CONTRASEÑA Y/O TIPO USUARIO ERRONEO',
        position: 'top-center',
        icon: 'error'
      });

    }
  },
  // error: OnErrorCall
});
```

Fuente: Elaboración propia

En la figura 85 se puede ver un fragmento del código de la vista de ingresar al sistema el cual se está enviando por Ajax mediante una ruta al controlador los atributos de usuario, contraseña y tipo para poder realizar la ejecución con la BD.

Figura 86 Controlador Ingresar al Sistema

```
@RequestMapping(value="/Login/{user},{pass},{tipo}", method = RequestMethod.POST, produces=MediaType.APPLICATION_JSON_VALUE)
public @ResponseBody Map<String, Object> getUsuario(HttpServletRequest request, HttpServletResponse response, @PathVariable String user, @PathVariable String pass, @PathVariable String tipo) throws Exception {
    Map<String, Object> mapResult = new HashMap<>();
    HttpSession session = request.getSession(true);
    try {
        UsuarioBean usuariosBean = usuariosService.AccesoUsuario(user, pass, tipo);
        mapResult.put(Constants.STATUS_TEXT, usuariosBean != null ? true : false);

        if (usuariosBean != null) {
            session.setAttribute("pusuarioBean", usuariosBean);
            mapResult.put("data", usuariosBean);
            mapResult.put("message", AvisosConstantes.AVISO_LOGIN_OK.replace("{0}", usuariosBean.getUsuario()));
        } else {
            mapResult.put("message", AvisosConstantes.AVISO_LOGIN_NO);
        }
    } catch (Exception ex) {
        ex.printStackTrace();
        mapResult.put(Constants.STATUS_TEXT, false);
        mapResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_LOGIN_NO);
        // TODO: handle exception
    }
    return mapResult;
}

// METODO POST LOGOUT USUARIO RESPONSEBODY
@RequestMapping(value="/Logout", method = RequestMethod.GET)
public void processRequestGET(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    String action=(request.getPathInfo()!=null?request.getPathInfo(): "");
    HttpSession session = request.getSession();

    session.invalidate();

    response.sendRedirect("../");
}
```

Fuente: Elaboración propia

En la figura 86 se puede ver el código del controlador usuario para poder realizar el envío a la BD para la validación de datos y obtener la respuesta de ingreso al sistema o mensaje de error.

Figura 87 Ejecución BD Ingresar al Sistema

```
@SuppressWarnings({ "rawtypes", "unchecked" })
@Override
public UsuarioBean AccesoUsuario(String user, String pass, String tipo) throws Exception {
    if(Log.isDebugEnabled()) Log.debug("Inicio - UsuariosDAOImpl.AccesoUsuario ");
    String SQL = "exec SP_TR_Acceso_Usuario ?,?,?";

    try {
        usuariosBean= (UsuarioBean)jdbcTemplate.queryForObject(SQL, new Object[] {user,pass,tipo},new RowMapper() {

            @Override
            public UsuarioBean mapRow(ResultSet rs, int rowNum) throws SQLException {
                UsuarioBean user = new UsuarioBean();
                user.setUsuario(rs.getString("usuario"));
                user.setTipo(rs.getString("tipo"));
                user.setNombreCompleto(rs.getString("nombrecompleto"));
                user.setUsu_tipDescri(rs.getString("descrtipo"));

                // TODO A1uto-generated method stub
                return user;
            }

        });
    } catch (Exception ex) {
        if(Log.isDebugEnabled()) Log.debug("Error - UsuariosDAOImpl.AccesoUsuario ");
        Log.error(ex,ex);
        throw ex;
        // TODO: handle exception
    } finally {
        if(Log.isDebugEnabled()) Log.debug("Final - UsuariosDAOImpl.AccesoUsuario ");
    }

    return usuariosBean;
}
```

Fuente: Elaboración propia

En la figura 87 se puede ver el código del modelo de datos para ingresar al sistema y su ejecución en la base de datos para luego devolver la respuesta.

Figura 88 Modelo Persona

```
package pe.edu.sanna.triage.modelo;
import com.fasterxml.jackson.annotation.JsonFormat;
public abstract class PersonaModel extends MedicoBean{

    @JsonFormat(shape = JsonFormat.Shape.STRING, pattern="yyyy-MM-dd", timezone = "GMT-5")
    int persona;
    String dni;
    int edad;
    String nombreCompleto;
    String fechaNacimiento;
    String sexo;
    String civil;
    String email;
    String direccion;
    String telefonono;

    public int getEdad() {
        return edad;
    }
    public void setEdad(int edad) {
        this.edad = edad;
    }
    public String getTelefonono() {
        return telefonono;
    }
    public void setTelefonono(String telefonono) {
        this.telefonono = telefonono;
    }
    int estado;
    public int getPersona() {
        return persona;
    }
    public void setPersona(int persona) {
        this.persona = persona;
    }
    public String getDni() {
        return dni;
    }
    public void setDni(String dni) {
        this.dni = dni;
    }
    public String getNombreCompleto() {
        return nombreCompleto;
    }
    public void setNombreCompleto(String nombreCompleto) {
        this.nombreCompleto = nombreCompleto;
    }
    public String getFechaNacimiento() {
        return fechaNacimiento;
    }
}
```

Fuente: Elaboración propia

En la figura 88 se puede ver el código del modelo persona, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase PersonaModel con un extends del modelo MedicoBean para poder realizar la herencia de atributos.

Figura 89 Vista Gestionar Usuario

```
$.ajax({
  type: "POST",
  url: "${pageContext.request.contextPath}/Usuario/Registrar",
  crossDomain: true,
  data: formToJSON,
  contentType: "application/json; charset=utf-8",
  dataType: "json",
  async: true,
  success: function (response) {
    //$("#modal-dialog").modal('hide');

    if(response.status === true){
      //alert(JSON.stringify(d));
      $.toast({
        heading: 'Exito',
        text: 'USUARIO REGISTRADO SATISFACTORIAMENTE',
        position: 'top-center',
        icon: 'success'
      });

      // alert("Usuario Registrado Correctamente");
      limpiarTxtUsuarios();
      $("#formManUsuarios")[0].reset();
      input1.disabled = true;
      input2.disabled = true;
      input3.disabled = true;
      input4.disabled = true;
      input5.disabled = true;
      input6.disabled = true;
      input7.disabled = true;
      input8.disabled = true;
      input9.disabled = true;
      input10.disabled = true;
      input11.disabled = true;
      input12.disabled = true;

      //CARGAR DATATABLE
      // LoadDataTableWithAjax()
      // $("#stack2").modal('hide');
      // $("#modal-dialog").modal('hide');
    }else{
      //alert("El DNI YA EXISTE: "+response.message)

      $.toast({
        heading: 'Warning',
        text: 'USUARIO YA REGISTRADO',
        position: 'top-center',
        showhideTransition: 'plain',
        icon: 'warning'
      });
    }
  }
});
```

Fuente: Elaboración propia

En la figura 89 se puede ver un fragmento del código de la vista de gestionar usuario el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 90 Controlador Gestionar Usuario

```
@RequestMapping(value="/Registrar",method=RequestMethod.POST,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getRegistrarUsuario(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        UsuarioBean usuarioBean = (UsuarioBean) new JsonSerializer().deserialize(jsonEnviado.toString(),UsuarioBean.class);

        result = usuariosService.registrarUsuario(usuarioBean);
        mapaResult.put(Constantes.STATUS_TEXT, result);
        mapaResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constantes.STATUS_TEXT, false);
        mapaResult.put(Constantes.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 90 se puede ver el código del controlador usuario para poder realizar el envío a la BD para la validación de datos y obtener la respuesta de registro de usuario o mensaje de error.

Figura 91 Ejecución BD Registrar Usuario

```
@Override
public boolean registrarUsuario(UsuarioBean usuarioBean) {
    // TODO Auto-generated method stub

    String SQL = "exec SP_TR_INSERT_USUARIOS ?,?,?,?,?,?,?,?,?,?,?,?,?,?";

    try {
        row = jdbcTemplate.update(SQL, new Object[] {
            usuarioBean.getDni(),
            usuarioBean.getNombreCompleto(),
            usuarioBean.getFechaNacimiento(),
            usuarioBean.getSexo(),
            usuarioBean.getCivil(),
            usuarioBean.getEmail(),
            usuarioBean.getDireccion(),
            usuarioBean.getEstado(),
            usuarioBean.getEspecialidad(),
            usuarioBean.getRne(),
            usuarioBean.getCmp(),
            usuarioBean.getTelefonono(),
            usuarioBean.getPass(),
            usuarioBean.getTipo(),
            usuarioBean.getUsuarioCreacion_use()
        });

    }

    catch (DataAccessException e) {
        e.printStackTrace();
        SQLException sqle = (SQLException) e.getCause();
        System.out.println("MENSAJE DEVOLUCION SQL"+sqle.getMessage());

        /*
        System.out.println("Error code: " + sqle.getErrorCode());
        System.out.println("SQL state: " + sqle.getSQLState());*/

        throw e;
    } catch (Exception e){
        e.printStackTrace();
        throw e;
    }

    return row==1;
}
```

Fuente: Elaboración propia

En la figura 91 se puede ver el código del modelo de datos para registrar usuario y su ejecución en la base de datos para luego devolver la respuesta.

Figura 92 Modelo Paciente

```
package pe.edu.sanna.triage.modelo;
import com.fasterxml.jackson.annotation.JsonFormat;

public abstract class PacienteModel extends PersonaBean{

    @JsonFormat(shape = JsonFormat.Shape.STRING, pattern="yyyy-MM-dd", timezone = "GMT-5")
    int pac_idpaciente;
    String pac_seguro;
    int pac_numseguro;
    String pac_raza;
    String pac_religion;
    String ocupacionPac;
    int pac_idpersona;
    String pac_usuarioCreacion;
    int pac_estado;

    public int getPac_idpaciente() {
        return pac_idpaciente;
    }
    public void setPac_idpaciente(int pac_idpaciente) {
        this.pac_idpaciente = pac_idpaciente;
    }

    public String getPac_seguro() {
        return pac_seguro;
    }
    public void setPac_seguro(String pac_seguro) {
        this.pac_seguro = pac_seguro;
    }
    public int getPac_numseguro() {
        return pac_numseguro;
    }
    public void setPac_numseguro(int pac_numseguro) {
        this.pac_numseguro = pac_numseguro;
    }

    public String getPac_raza() {
        return pac_raza;
    }
    public void setPac_raza(String pac_raza) {
        this.pac_raza = pac_raza;
    }
    public String getPac_religion() {
        return pac_religion;
    }
    public void setPac_religion(String pac_religion) {
        this.pac_religion = pac_religion;
    }
    public int getPac_idpersona() {
        return pac_idpersona;
    }
}
```

Fuente: Elaboración propia

En la figura 92 se puede ver el código del modelo paciente, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase PacienteModel con un extends del modelo PersonaBean para poder realizar la herencia de atributos.

Figura 93 Vista Gestionar Paciente

```

$.ajax({
    type: "POST",
    url: "${pageContext.request.contextPath}/Paciente/Registrar",
    crossDomain: true,
    data: formToJSON(),
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    success: function (response) {
        //$("#modal-dialog").modal('hide');
        if(response.status === true){
            $.toast({
                heading: 'Exito',
                text: 'PACIENTE REGISTRADO SATISFACTORIAMENTE',
                position: 'top-center',
                icon: 'success'
            });

            //alert("Paciente Registrado Correctamente");
            limpiarTxtPacientes();
            $("#formPantPacientes")[0].reset();
            //CARCAR DATATABLE
            // loadDataTableWithAjax()
            // $("#stack2").modal('hide');
            // $("#modal-dialog").modal('hide');
        }else{
            $.toast({
                heading: 'Error',
                text: 'PACIENTE YA REGISTRADO',
                position: 'top-center',
                icon: 'error'
            });

            //alert("No Se Pudo Registrar")
        }
    },
    error: onErrorCall
});
}
}

```

Fuente: Elaboración propia

En la figura 93 se puede ver un fragmento del código de la vista de gestionar paciente el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 94 Controlador Gestionar Paciente

```

@RequestMapping(value="/Registrar",method=RequestMethod.POST,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getRegistrarUsuario(HttpServletRequest request, HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        PacienteBean pacienteBean = (PacienteBean) new JsonSerializer().deserialize(jsonEnviado.toString(),PacienteBean.class);

        result = pacienteService.registrarPaciente(pacienteBean);
        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));

    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}

```

Fuente: Elaboración propia

En la figura 94 se puede ver el código del controlador paciente para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de registro de paciente o mensaje de error.

Figura 95 Ejecución BD Registrar Paciente

```
@Override
public boolean insertarPaciente(PacienteBean Paciente) throws Exception {
    // TODO Auto-generated method stub
    if(log.isDebugEnabled()) log.debug("Inicio - PacienteDAOImpl.insertarPaciente");
    String SQL = "exec SP_TR_INSERTPACIENTE ?,?,?,?,?,?,?,?,?,?,?,?,?,?";
    try {
        rows= jdbcTemplate.update(SQL.toString(),new Object[] {
            Paciente.getDni(),
            Paciente.getSexo(),
            Paciente.getFechaNacimiento(),
            Paciente.getNombreCompleto(),
            Paciente.getDireccion(),
            Paciente.getCivil(),
            Paciente.getOcupacionPac(),
            Paciente.getPac_raza(),
            Paciente.getPac_religion(),
            Paciente.getEmail(),
            Paciente.getPac_estado(),
            Paciente.getTelefonono(),
            Paciente.getPac_usuarioCreacion(),
            Paciente.getPac_seguro()
        });
    } catch (DataAccessException e) {
        e.printStackTrace();
        SQLException sqle = (SQLException) e.getCause();
        System.out.println("MENSAJE DEVOLUCION SQL"+sqle.getMessage());

        /*
        System.out.println("Error code: " + sqle.getErrorCode());
        System.out.println("SQL state: " + sqle.getSQLState());*/

        throw e;
    } catch (Exception e){
        e.printStackTrace();
        throw e;
    }

    return rows--1;
}
```

Fuente: Elaboración propia

En la figura 95 se puede ver el código del modelo de datos para registrar paciente y su ejecución en la base de datos para luego devolver la respuesta.

Figura 96 Modelo Orden Atención

```
import com.fasterxml.jackson.annotation.JsonFormat;[]
public abstract class OrdenAtencionModel extends PacienteBean {

    @JsonFormat(shape = JsonFormat.Shape.STRING, pattern="yyyy-MM-dd", timezone = "GMT-5")

    int idordenAtencion;
    int idordentabla;
    int codic;
    String prioridad;
    String nombreCompleto;
    String sexo;
    int edad;
    String fechaHora;
    String hora;
    String estado_triaje;
    int idTurno;
    String idEmpresaAsegura;
    String usuarioCreacionAte;
    String lilegPaciente;
    String detilegPaciente;
    String estadoAtenEmergencia;
    String estado_enfermeria;
    String fecha_apertura_triaje;
    String fecha_apertura_emergencia;
    String usuario_regenfermeria;
    String topico;
    String usuarioalta;
    String pass;
    String observacion_alta;
    int horadiiff;
    String archivo;

    public String getArchivo() {
        return archivo;
    }
    public void setArchivo(String archivo) {
        this.archivo = archivo;
    }
    public int getIdordentabla() {
        return idordentabla;
    }
    public void setIdordentabla(int idordentabla) {
        this.idordentabla = idordentabla;
    }
    public int getHoradiiff() {
        return horadiiff;
    }
    public void setHoradiiff(int horadiiff) {
        this.horadiiff = horadiiff;
    }
}
```

Fuente: Elaboración propia

En la figura 96 se puede ver el código del modelo orden atención, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase OrdenAtencionModel con un extends del modelo PacienteBean para poder realizar la herencia de atributos.

Figura 97 Vista Crear Orden Atención

```
$.ajax({
  type: "POST",
  url: "${pageContext.request.contextPath}/OrdenAtencion/Registrar",
  crossDomain: true,
  data: formToJSON,
  contentType: "application/json; charset=utf-8",
  dataType: "json",
  async: true,
  success: function (response) {
    if(response.status === true){
      $.toast({
        heading: 'Exito',
        text: 'ORDEN ATENCION CREADA SATISFACTORIAMENTE',
        position: 'top-center',
        icon: 'success'
      });
      limpiarTxtOrdenes();
      //CARGAR DATATABLE
      // LoadDataTableWithAjax()
      // $("#stack2").modal('hide');
      // $("#modal-dialog").modal('hide');
    }else{
      $.toast({
        heading: 'error',
        text: 'NO SE PUDO CREAR ORDEN ATENCION',
        position: 'top-center',
        icon: 'error'
      });
    }
  }
},
//error: OnErrorCall
);
});
```

Fuente: Elaboración propia

En la figura 97 se puede ver un fragmento del código de la vista de crear orden atención el cual se está enviando por Ajax mediante una ruta al controlador los atributos a crear para poder realizar la ejecución con la BD.

Figura 98 Controlador Crear Orden Atención

```
@RequestMapping(value="/Registrar",method=RequestMethod.POST,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getRegistrarUsuario(HttpServletRequest request, HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        OrdenAtencionBean ordenAtencionBean = (OrdenAtencionBean) new JsonSerializer().deserialize(jsonEnviado.toString(),OrdenAtencionBean.class);

        result = ordenAtencionService.registrarOrdenAtencion(ordenAtencionBean);
        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));

    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 98 se puede ver el código del controlador orden atención para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de la creación orden atención o mensaje de error.

Figura 99 Ejecución BD Crear Orden Atención

```

@Override
public boolean registrarOrdenAtencion(OrdenAtencionBean ordenAtencionBean) throws Exception {
    // TODO Auto-generated method stub
    if(Log.isDebugEnabled()) Log.debug("Inicio - OrdenAtencionDAOImpl.registrarOrdenAtencion");
    String SQL="exec SP_TR_INSERTAR_ORDEN_ATENCION ?,?,?,?,?,?,?,?,?,?";
    try {
        rows=jdbcTemplate.update(SQL,new Object[] {

            ordenAtencionBean.getPac_idpaciente(),
            ordenAtencionBean.getFechaHora(),
            ordenAtencionBean.getEstado_triaje(),
            ordenAtencionBean.getHora(),
            //ordenAtencionBean.getIdTurno(),
            ordenAtencionBean.getIdEmpresaAsegura(),
            ordenAtencionBean.getIdUsuarioCreacionAtc(),
            ordenAtencionBean.getIdLegPaciente(),
            ordenAtencionBean.getDetlLegPaciente(),
            ordenAtencionBean.getEstado_enfermeria(),
            ordenAtencionBean.getFecha_apertura_triaje(),
            ordenAtencionBean.getFecha_apertura_emergencia()

        });
    } catch (Exception ex) {
        // TODO: handle exception
        if(Log.isDebugEnabled()) Log.debug("Error - OrdenAtencionDAOImpl.registrarOrdenAtencion");
        Log.error(ex,ex);
        throw ex;
    }finally{
        if (Log.isDebugEnabled()) Log.debug("Final - OrdenAtencionDAOImpl.registrarOrdenAtencion");
    }
    return rows--1;
}

```

Fuente: Elaboración propia

En la figura 99 se puede ver el código del modelo de datos para crear orden atención y su ejecución en la base de datos para luego devolver la respuesta.

Figura 100 Vista Listar Orden Atención

```
function LoadDataTableWithAjax(){
    var fechaInicio = $('#txtFechaInicio').val()
    var fechaFin = $('#txtFechaFin').val()
    validarfecha();

    $('#data-table-default').dataTable().fnClearTable();
    $('#data-table-default').dataTable().fnDraw();

    $.ajax({
        type: "GET",
        url: "${pageContext.request.contextPath}/OrdenAtencion/Rest/" + fechaInicio + "," + fechaFin,
        crossDomain: true,
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        async: true,
        success: function (response) {
            //ADD NOTIFICACION
            //ShowMessage(response.message);

            if(response.data.length > 0){
                //UPDATE DATATABLES
                $('#data-table-default').dataTable().fnAddData(response.data);
            }else{
                console.log("No hay Atenciones")
            }
        }
    })
}

$( "#btnBuscar" ).click(function() {
    // $('#modal-dialog').modal('show');
    //$('#stack1').modal('show')
    LoadDataTableWithAjax()
});
```

Fuente: Elaboración propia

En la figura 100 se puede ver un fragmento del código de las vistas de bandeja admisionista, internista, médico y enfermera el cual se está enviando por Ajax mediante una ruta al controlador los atributos a listar para poder realizar la ejecución con la BD.

Figura 101 Controlador Listar Orden Atención

```
@RequestMapping(value="/Rest/{fechaInicio},{fechaFin}",method=RequestMethod.GET,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getOrdenAtencion(@PathVariable String fechaInicio, @PathVariable String fechaFin) throws Exception{

    Map<String, Object>mapResult=new HashMap<>();

    try {
        List<OrdenAtencionBean> lstOrden =ordenAtencionService.listarOrdenAtencion(fechaInicio,fechaFin);
        mapResult.put(Constantes.STATUS_TEXT, true);
        mapResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_READ.replace("{0}", String.valueOf(lstOrden.size())));
        mapResult.put("data", lstOrden);
    } catch (Exception e) {
        Log.error("OrdenAtencionController.getOrdenes - Error");
        Log.error(e.getMessage());
        mapResult.put(Constantes.STATUS_TEXT, false);
        mapResult.put(Constantes.MESSAGE_TEXT, e.getLocalizedMessage());
        // TODO: handle exception
    }
    return mapResult;
}
```

Fuente: Elaboración propia

En la figura 101 se puede ver el código del controlador orden atención para poder realizar el envío a la BD para la validación de datos y obtener la respuesta de listar orden atención o mensaje de error.

Figura 102 Ejecución BD Listar Orden Atención

```
@Override
public List<OrdenAtencionBean> listarOrdenAtencion(String fechaInicio, String fechaFin) throws Exception {
    // TODO Auto-generated method stub
    if (Log.isDebugEnabled())
        Log.debug("Inicio - OrdenAtencionDAOImpl.listarOrdenAtencion");
    List<OrdenAtencionBean> lstOrden;
    String SQL = "exec SP_LISTAR_ORDEN_ATENCION ?,?";
    try {
        lstOrden = jdbcTemplate.query(SQL, new Object[] { fechaInicio, fechaFin }, new ResultSetExtractor<List<OrdenAtencionBean>>() {
            @Override
            public List<OrdenAtencionBean> extractData(ResultSet rs) throws SQLException, DataAccessException {
                // TODO Auto-generated method stub
                List<OrdenAtencionBean> lista = new ArrayList<>();
                while (rs.next()) {
                    OrdenAtencionBean orden= new OrdenAtencionBean();
                    orden.setIdordenAtencion(rs.getInt("idordenatencion"));
                    orden.setCodHc(rs.getInt("codhc"));
                    orden.setNombreCompleto(rs.getString("nombrecompleto"));
                    orden.setSexo(rs.getString("sexo"));
                    orden.setEdad(rs.getInt("edad"));
                    orden.setFechaHora(rs.getString("fechahora"));
                    orden.setHora(rs.getString("hora"));
                    orden.setEstado_triaje(rs.getString("estado_triaje"));

                    lista.add(orden);
                }
                return lista;
            }
        });
    } catch (Exception ex) {
        // TODO: handle exception
        if (Log.isDebugEnabled()) Log.debug("Error - OrdenAtencionDAOImpl.listarOrdenAtencion");
        Log.error(ex, ex);
        throw ex;
    } finally {
        if (Log.isDebugEnabled()) Log.debug("Final - OrdenAtencionDAOImpl.listarOrdenAtencion");
    }
    return lstOrden;
}
```

Fuente: Elaboración propia

En la figura 102 se puede ver el código del modelo de datos para listar orden atención y su ejecución en la base de datos para luego devolver la respuesta.

Figura 103 Modelo Prioridad

```
package pe.edu.sanna.triage.model;

public class PrioridadesModel {

    int idDiagnostico;
    String discriminador1;
    String discriminador2;
    String discriminador3;
    String discriminador4;
    String discriminador5;
    String discriminador6;
    String discriminador7;
    String discriminador8;
    String discriminador9;
    String discriminador10;

    String discriminador11;
    String discriminador12;
    String discriminador13;
    String discriminador14;
    String discriminador15;
    String discriminador16;
    String discriminador17;
    String discriminador18;
    String discriminador19;
    String discriminador20;

    String discriminador21;
    String discriminador22;
    String discriminador23;
    String discriminador24;
    String discriminador25;
    String discriminador26;
    String discriminador27;
    String discriminador28;
    String discriminador29;
    String discriminador30;

    String discriminador31;
    String discriminador32;
    String discriminador33;
    String discriminador34;
    String discriminador35;
    String discriminador36;
    String discriminador37;
    String discriminador38;
    String discriminador39;
    String discriminador40;
    String discriminador41;

    public int getIdDiagnostico() {
        return idDiagnostico;
    }

    public void setIdDiagnostico(int idDiagnostico) {
        this.idDiagnostico = idDiagnostico;
    }
}
```

Fuente: Elaboración propia

En la figura 103 se puede ver el código del modelo prioridad, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase PrioridadModel.

Figura 105 Vista Motivo Atención

```
function lol(value){
    var name = $('#txtAbusoMa').val()
    var txtAdulMal=$('#txtAdulMal').val()
    var txtAdultosinlip=$('#txtAdultosinlip').val()
    var txtAgresion=$('#txtAgresion').val()
    var txtAlerg=$('#txtAlerg').val()
    var txtEbrrio=$('#txtEbrrio').val()

    var txtAsma=$('#txtAsma').val()
    var txtAutolesion=$('#txtAutolesion').val()
    var txtBebeMal=$('#txtBebeMal').val()
    var txtCaidas=$('#txtCaidas').val()
    var txtCefalea=$('#txtCefalea').val()
    var txtComext=$('#txtComext').val()
    var txtConvulsiones=$('#txtConvulsiones').val()
    var txtCuerpoExt=$('#txtCuerpoExt').val()
    var txtDiabetes=$('#txtDiabetes').val()
    var txtDiavom=$('#txtDiavom').val()

    var txtDisneaA=$('#txtDisneaA').val()
    var txtDisneaN=$('#txtDisneaN').val()
    var txtDolorAbA=$('#txtDolorAbA').val()
    var txtDolorAbN=$('#txtDolorAbN').val()
    var txtDolorCuello=$('#txtDolorCuello').val()
    var txtDolorEsp=$('#txtDolorEsp').val()
    var txtDolorGar=$('#txtDolorGar').val()
    var txtFesticular=$('#txtFesticular').val()
    var txtDolorForacico=$('#txtDolorForacico').val()
    var txtEmbarazo=$('#txtEmbarazo').val()

    var txtETS=$('#txtETS').val()
    var txtEnfMental=$('#txtEnfMental').val()
    var txtExantemas=$('#txtExantemas').val()
    var txtExQuimica=$('#txtExQuimica').val()
    var txtHemoGatro=$('#txtHemoGatro').val()
    var txtHemoVaginal=$('#txtHemoVaginal').val()
    var txtHerida=$('#txtHerida').val()
    var txtInfLocAbs=$('#txtInfLocAbs').val()
    var txtLesionesTronco=$('#txtLesionesTronco').val()
    var txtMordedura=$('#txtMordedura').val()

    var txtNeomalest=$('#txtNeomalest').val()
    var txtNinoCojenado=$('#txtNinoCojenado').val()
    var txtNinomalest=$('#txtNinomalest').val()
    var txtNinoIrritable=$('#txtNinoIrritable').val()
    var txtPadrePreo=$('#txtPadrePreo').val()
    var txtPalpitaciones=$('#txtPalpitaciones').val()
    var txtPolitraumatismo=$('#txtPolitraumatismo').val()
    var txtProblemaDental=$('#txtProblemaDental').val()
    var txtProblemaExtremo=$('#txtProblemaExtremo').val()
    var txtProblemaFacial=$('#txtProblemaFacial').val()
    var txtProblemaOcular=$('#txtProblemaOcular').val()
}
```

Fuente: Elaboración propia

Figura 104 Vista Listar Motivo Atención

```
$.ajax({
    type: "GET",
    url: "RestPrioridad/" + value,
    crossDomain: true,

    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    success: function (response2) {
        if(response2.status === true){
            var d = response2.data;

            //alert(JSON.stringify(d));
            //alert(d);

            discr11 = d.disciminador1;
            discr12 = d.disciminador2;
            discr13 = d.disciminador3;
            discr14 = d.disciminador4;
            discr15 = d.disciminador5;
            discr16 = d.disciminador6;
            discr17 = d.disciminador7;
            discr18 = d.disciminador8;
            discr19 = d.disciminador9;
            discr110 = d.disciminador10;
            discr111 = d.disciminador11;
            discr112 = d.disciminador12;
            discr113 = d.disciminador13;
            discr114 = d.disciminador14;

            discr115 = d.disciminador15;
            discr116 = d.disciminador16;
            discr117 = d.disciminador17;
            discr118 = d.disciminador18;
            discr119 = d.disciminador19;
            discr120 = d.disciminador20;
            discr121 = d.disciminador21;
            discr122 = d.disciminador22;
            discr123 = d.disciminador23;
            discr124 = d.disciminador24;
            discr125 = d.disciminador25;

            discr126 = d.disciminador26;
            discr127 = d.disciminador27;
            discr128 = d.disciminador28;
            discr129 = d.disciminador29;
            discr130 = d.disciminador30;
            discr131 = d.disciminador31;
            discr132 = d.disciminador32;
            discr133 = d.disciminador33;
            discr134 = d.disciminador34;
            discr135 = d.disciminador35;
            discr136 = d.disciminador36;
            discr137 = d.disciminador37;
        }
    }
});
```

Fuente: Elaboración propia

Figura 106 Vista Listar Motivo Atención

```

        disci40 = d.disciminador40;
        disci41 = d.disciminador41;

        $('#Discriminador1').text($.trim(d.disciminador1));
        $('#Discriminador2').text($.trim(d.disciminador2));
        $('#Discriminador3').text($.trim(d.disciminador3));
        $('#Discriminador4').text($.trim(d.disciminador4));
        $('#Discriminador5').text($.trim(d.disciminador5));
        $('#Discriminador6').text($.trim(d.disciminador6));
        $('#Discriminador7').text($.trim(d.disciminador7));
        $('#Discriminador8').text($.trim(d.disciminador8));
        $('#Discriminador9').text($.trim(d.disciminador9));
        $('#Discriminador10').text($.trim(d.disciminador10));

        $('#Discriminador11').text($.trim(d.disciminador11));
        $('#Discriminador12').text($.trim(d.disciminador12));
        $('#Discriminador13').text($.trim(d.disciminador13));
        $('#Discriminador14').text($.trim(d.disciminador14));
        $('#Discriminador15').text($.trim(d.disciminador15));
        $('#Discriminador16').text($.trim(d.disciminador16));
        $('#Discriminador17').text($.trim(d.disciminador17));
        $('#Discriminador18').text($.trim(d.disciminador18));
        $('#Discriminador19').text($.trim(d.disciminador19));
        $('#Discriminador20').text($.trim(d.disciminador20));

        $('#Discriminador21').text($.trim(d.disciminador21));
        $('#Discriminador22').text($.trim(d.disciminador22));
        $('#Discriminador23').text($.trim(d.disciminador23));
        $('#Discriminador24').text($.trim(d.disciminador24));
        $('#Discriminador25').text($.trim(d.disciminador25));
        $('#Discriminador26').text($.trim(d.disciminador26));
        $('#Discriminador27').text($.trim(d.disciminador27));
        $('#Discriminador28').text($.trim(d.disciminador28));
        $('#Discriminador29').text($.trim(d.disciminador29));
        $('#Discriminador30').text($.trim(d.disciminador30));

        $('#Discriminador31').text($.trim(d.disciminador31));
        $('#Discriminador32').text($.trim(d.disciminador32));
        $('#Discriminador33').text($.trim(d.disciminador33));
        $('#Discriminador34').text($.trim(d.disciminador34));
        $('#Discriminador35').text($.trim(d.disciminador35));
        $('#Discriminador36').text($.trim(d.disciminador36));
        $('#Discriminador37').text($.trim(d.disciminador37));
        $('#Discriminador38').text($.trim(d.disciminador38));
        $('#Discriminador39').text($.trim(d.disciminador39));
        $('#Discriminador40').text($.trim(d.disciminador40));
        $('#Discriminador41').text($.trim(d.disciminador41));

    }
    }
    combosGenericos();
    },
    //error: OnErrorCall
});

if(value=='1'){
    $('#txtMotivoIngreso').text(name)
}

```

Fuente: Elaboración propia

En las figuras 104, 105, 106 se puede ver unos fragmentos del código de la vista de registrar triage el cual se está enviando por Ajax mediante una ruta al controlador los atributos a listar los motivos de atención y discriminadores para poder realizar la ejecución con la BD.

Figura 107 Controlador Listar Motivo

```

@RequestMapping(value="/RestPrioridad/{codigo}",
    method = RequestMethod.GET,
    produces={"application/json","application/xml"})
public @ResponseBody Map<String, Object> getPersona(@PathVariable String codigo) throws Exception {
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        // LISTAR X CODIGO
        PrioridadBean pBean =prioridadService.listarDiscriminadorxCod(codigo);
        mapaResult.put(Constants.STATUS_TEXT, (pBean != null? true : false));
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_READ.replace("{0}",(pBean != null? "1":"0")));

        if(pBean != null) {
            mapaResult.put("data", pBean);
        }
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}

```

Fuente: Elaboración propia

En la figura 107 se puede ver el código del controlador triage para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de listar motivo atención o mensaje de error.

Figura 108 Ejecución BD Listar Motivo Atención

```

public PrioridadBean listarDiscriminadorxCod(String codigo) throws Exception {
    // TODO Auto-generated method stub
    if(Log.isDebugEnabled()) Log.debug("Inicio - PrioridadDAOImpl.listarDiscriminadorxCod");
    String SQL = "EXEC SP_DISCRIMINADORES ?";

    try {
        prioridadBean=(PrioridadBean)jdbcTemplate.queryForObject(SQL,new Object[] {codigo},new RowMapper() {
            @Override
            public Object mapRow(ResultSet rs, int rowNum) throws SQLException {
                PrioridadBean pBean = new PrioridadBean();
                pBean.setIdDagnostico(rs.getInt("idDagnostico"));
                pBean.setDiscriminador1(rs.getString("Discriminador1"));
                pBean.setDiscriminador2(rs.getString("Discriminador2"));
                pBean.setDiscriminador3(rs.getString("Discriminador3"));
                pBean.setDiscriminador4(rs.getString("Discriminador4"));
                pBean.setDiscriminador5(rs.getString("Discriminador5"));
                pBean.setDiscriminador6(rs.getString("Discriminador6"));
                pBean.setDiscriminador7(rs.getString("Discriminador7"));
                pBean.setDiscriminador8(rs.getString("Discriminador8"));
                pBean.setDiscriminador9(rs.getString("Discriminador9"));
                pBean.setDiscriminador10(rs.getString("Discriminador10"));

                pBean.setDiscriminador11(rs.getString("Discriminador11"));
                pBean.setDiscriminador12(rs.getString("Discriminador12"));
                pBean.setDiscriminador13(rs.getString("Discriminador13"));
                pBean.setDiscriminador14(rs.getString("Discriminador14"));
                pBean.setDiscriminador15(rs.getString("Discriminador15"));
                pBean.setDiscriminador16(rs.getString("Discriminador16"));
                pBean.setDiscriminador17(rs.getString("Discriminador17"));
                pBean.setDiscriminador18(rs.getString("Discriminador18"));
                pBean.setDiscriminador19(rs.getString("Discriminador19"));
                pBean.setDiscriminador20(rs.getString("Discriminador20"));

                pBean.setDiscriminador21(rs.getString("Discriminador21"));
                pBean.setDiscriminador22(rs.getString("Discriminador22"));
                pBean.setDiscriminador23(rs.getString("Discriminador23"));
                pBean.setDiscriminador24(rs.getString("Discriminador24"));
                pBean.setDiscriminador25(rs.getString("Discriminador25"));
                pBean.setDiscriminador26(rs.getString("Discriminador26"));
                pBean.setDiscriminador27(rs.getString("Discriminador27"));
                pBean.setDiscriminador28(rs.getString("Discriminador28"));
                pBean.setDiscriminador29(rs.getString("Discriminador29"));
                pBean.setDiscriminador30(rs.getString("Discriminador30"));

                pBean.setDiscriminador31(rs.getString("Discriminador31"));
                pBean.setDiscriminador32(rs.getString("Discriminador32"));
                pBean.setDiscriminador33(rs.getString("Discriminador33"));
                pBean.setDiscriminador34(rs.getString("Discriminador34"));
                pBean.setDiscriminador35(rs.getString("Discriminador35"));
                pBean.setDiscriminador36(rs.getString("Discriminador36"));
                pBean.setDiscriminador37(rs.getString("Discriminador37"));
                pBean.setDiscriminador38(rs.getString("Discriminador38"));
            }
        });
    }
}

```

Fuente: Elaboración propia

En la figura 108 se puede ver el código del modelo de datos para listar motivo atención y discriminadores y su ejecución en la base de datos para luego devolver la respuesta.

Figura 109 Modelo Triage

```

package pe.edu.sanna.triage.modelo;

public abstract class TriageModel {

    int idordenAtencion;
    String diagnostico;
    String fechaAtencion;
    String motivo;
    String antecedentes;
    String alergias;
    String peso;
    String pa;
    String pa2;
    String temperatura;
    String sato2;
    String fc;
    String fr;
    String fur;
    String observacion;
    String prioridad;
    String topico;
    String usu_creacionTriage;

    public String getUsu_creacionTriage() {
        return usu_creacionTriage;
    }
    public void setUsu_creacionTriage(String usu_creacionTriage) {
        this.usu_creacionTriage = usu_creacionTriage;
    }
    public int getIdordenAtencion() {
        return idordenAtencion;
    }
    public void setIdordenAtencion(int idordenAtencion) {
        this.idordenAtencion = idordenAtencion;
    }
    public String getDiagnostico() {
        return diagnostico;
    }
    public void setDiagnostico(String diagnostico) {
        this.diagnostico = diagnostico;
    }
    public String getFechaAtencion() {
        return fechaAtencion;
    }
    public void setFechaAtencion(String fechaAtencion) {
        this.fechaAtencion = fechaAtencion;
    }
    public String getMotivo() {
        return motivo;
    }
    public void setMotivo(String motivo) {
        this.motivo = motivo;
    }
}

```

Fuente: Elaboración propia

En la figura 109 se puede ver el código del modelo triage, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase TriageModel.

Figura 110 Vista Registrar Triage

```

$.ajax({
    type: "POST",
    url: "${pageContext.request.contextPath}/Triage/Rest",
    crossDomain: true,
    data: formToJSON,
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    success: function (response) {
        $("#modal-dialog").modal('hide');
        if(response.status === true){
            $.ajax({
                type: "POST",
                url: "${pageContext.request.contextPath}/Triage/RestDetalle",
                crossDomain: true,
                data: dataToJSON,
                contentType: "application/json; charset=utf-8",
                dataType: "json",
                async: true,
                success: function (response) {
                    if(response.status === true){
                        //error: OnErrorCall
                    }
                }
            });
            //CARGAR DATATABLE
            $.toast({
                heading: 'Exito',
                text: 'TRIAJE REGISTRADO SATISFACTORIAMENTE',
                position: 'top-center',
                icon: 'success'
            });
            LoadDataTableWithAjax();
            $("#stack2").modal('hide');
            $("#modal-dialog").modal('hide');
            setTimeout('window.location.reload()',2000);
            // window.location.reload();
        }else{
            $.toast({
                heading: 'Error',
                text: 'NO SE PUDO REGISTRAR TRIAJE',
                position: 'top-center'
            });
        }
    }
});

```

Fuente: Elaboración propia

En la figura 110 se puede ver un fragmento del código de la vista de registrar triage el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 111 Controlador Triage

```
@RequestMapping(value="/Rest",method=RequestMethod.POST,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getRegistrarTriage(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        TriageBean triageBean = (TriageBean) new JsonSerializer().deserialize(jsonEnviado.toString(),TriageBean.class);
        result = service.registrarTriage(triageBean);
        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}

@RequestMapping(value="/RestDetalle",method=RequestMethod.POST,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getRegistrarDetalleTriage(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        DetalleTriageBean triageBean = (DetalleTriageBean) new JsonSerializer().deserialize(jsonEnviado.toString(),DetalleTriageBean.class);
        result = service.registrarDetalleTriage(triageBean);
        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
}
```

Fuente: Elaboración propia

En la figura 111 se puede ver el código del controlador triage para poder realizar el envío a la BD para la validación de datos y obtener la respuesta de registro de triage o mensaje de error.

Figura 112 Ejecución BD Registrar Triage

```
@Override
public boolean registrarTriage(TriageBean triageBean) throws Exception {
    // TODO Auto-generated method stub
    if ((log.isDebugEnabled()) log.debug("Inicio - TriageDAOImpl.registrarTriage");
    String SQL = "exec SP_GUARDAR_TRIAGE ?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?";

    try {
        row = jdbcTemplate.update(SQL,new Object[] {
            triageBean.getIdordenAtencion(),
            //triageBean.getDiagnostico(),
            triageBean.getFechaAtencion(),
            //triageBean.getMotivo(),
            triageBean.getAntecedentes(),
            triageBean.getAlengias(),
            triageBean.getPeso(),
            triageBean.getPa(),
            triageBean.getPa2(),
            triageBean.getTemperatura(),
            triageBean.getSato2(),
            triageBean.getFc(),
            triageBean.getFr(),
            triageBean.getFur(),
            triageBean.getObservacion(),
            triageBean.getPrioridad(),
            triageBean.getTopico(),
            triageBean.getUsu_creacionTriage()
        });
    } catch (Exception ex) {
        // TODO: handle exception
        if ((log.isDebugEnabled()) log.debug("Error - TriageDAOImpl.registrarTriage");
        log.error(ex, ex);
        throw ex;
    }finally {
        if ((log.isDebugEnabled()) log.debug("Final - TriageDAOImpl.registrarTriage");
    }
    return row==1;
}
```

Fuente: Elaboración propia

En la figura 112 se puede ver el código del modelo de datos para registrar triage y su ejecución en la base de datos para luego devolver la respuesta.

Figura 113 Ejecución Registrar Triage

```

@Override
public boolean registrarDetalleTriajeBean(TriajeBean throws Exception {
    // TODO Auto-generated method stub

    if (Log.isDebugEnabled()) Log.debug("Inicio - TriajeDAOImpl.registrarDetalleTriaje");
    String SQL = "EXEC DETALLE_EMER_TRIAJE ?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?";

    try {

        row = jdbcTemplate.update(SQL,new Object[] {

            triajeBean.getIdordenAtencion(),
            triajeBean.getDiagnostico(),
            triajeBean.getDiscriminador1(),
            triajeBean.getDiscriminador2(),
            triajeBean.getDiscriminador3(),
            triajeBean.getDiscriminador4(),
            triajeBean.getDiscriminador5(),
            triajeBean.getDiscriminador6(),
            triajeBean.getDiscriminador7(),
            triajeBean.getDiscriminador8(),
            triajeBean.getDiscriminador9(),
            triajeBean.getDiscriminador10(),
            triajeBean.getDiscriminador11(),
            triajeBean.getDiscriminador12(),
            triajeBean.getDiscriminador13(),
            triajeBean.getDiscriminador14(),
            triajeBean.getDiscriminador15(),
            triajeBean.getDiscriminador16(),
            triajeBean.getDiscriminador17(),
            triajeBean.getDiscriminador18(),
            triajeBean.getDiscriminador19(),
            triajeBean.getDiscriminador20(),

            triajeBean.getDiscriminador21(),
            triajeBean.getDiscriminador22(),
            triajeBean.getDiscriminador23(),
            triajeBean.getDiscriminador24(),
            triajeBean.getDiscriminador25(),
            triajeBean.getDiscriminador26(),
            triajeBean.getDiscriminador27(),
            triajeBean.getDiscriminador28(),
            triajeBean.getDiscriminador29(),
            triajeBean.getDiscriminador30(),
            triajeBean.getDiscriminador31(),
            triajeBean.getDiscriminador32(),
            triajeBean.getDiscriminador33(),
            triajeBean.getDiscriminador34(),
            triajeBean.getDiscriminador35(),
            triajeBean.getDiscriminador36(),
            triajeBean.getDiscriminador37(),
            triajeBean.getDiscriminador38(),
            triajeBean.getDiscriminador39(),
            triajeBean.getDiscriminador40()

        });
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

```

Fuente: Elaboración propia

En la figura 113 se puede ver el código del modelo de datos para registrar triage y su ejecución en la base de datos para luego devolver la respuesta.

Figura 114 Vista Abandono

```

$( "#btnAbandonar" ).click(function() {
    //alert("fff");

    //alert(formToJSONIDORDE);

    $.ajax({
        type: "PUT",
        url: "${pageContext.request.contextPath}/OrdenAtencion/Abandono",
        crossDomain: true,
        data: formToJSONIDORDE,
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        async: true,
        success: function (response) {
            if(response.status === true){
                $.toast({
                    heading: 'Exito',
                    text: 'ABANDONO ACTUALIZADO SATISFACTORIAMENTE',
                    position: 'top-center',
                    icon: 'success'
                });
                LoadDataTableWithAjax();
            }
            // } else{
            //     $.toast({
            //         heading: 'Warning',
            //         text: 'SELECCIONE FILA PARA ACTUALIZAR ABANDONO',
            //         position: 'top-center',
            //         showHideTransition: 'plain',
            //         icon: 'warning'
            //     });
            // }

        });
    });
});

```

Fuente: Elaboración propia

En la figura 114 se puede ver un fragmento del código de la vista de registrar triage el cual se está enviando por Ajax mediante una ruta al controlador los atributos a actualizar estado a abandono para poder realizar la ejecución con la BD.

Figura 115 Controlador Abandono

```
@RequestMapping(value="/Abandono",method=RequestMethod.PUT,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getActualizarEstadoAbanTriage(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }

        OrdenAtencionBean orden = (OrdenAtencionBean) new JsonSerializer().deserialize(jsonEnviado.toString(), OrdenAtencionBean.class);

        // ACTUALIZAR USUARIO
        result = ordenAtencionService.actualizarEstadoAbandonoTriage(orden);

        mapaResult.put(Constantes.STATUS_TEXT, result);
        mapaResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_UPDATE.replace("{0}", (result ? "1": "0")));

    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constantes.STATUS_TEXT, false);
        mapaResult.put(Constantes.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 115 se puede ver el código del controlador orden atención para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de actualización de estado de abandono o mensaje de error.

Figura 116 Ejecución BD Abandono

```
@Override
public boolean actualizarEstadoAbandonoTriage(OrdenAtencionBean ordenAtencionBean) throws Exception {
    // TODO Auto-generated method stub
    if (Log.isDebugEnabled()) Log.debug("Inicio - OrdenAtencionDAOImpl.actualizarEstadoAbandonoTriage");

    String SQL = "exec SP_ACTUALIZAR_ESTADO_ABANDONO ?,?";

    try {
        rows = jdbcTemplate.update(SQL, new Object[] {
            ordenAtencionBean.getIdordenAtencion(),
            ordenAtencionBean.getUsuarioalta()
        });
    } catch (Exception ex) {
        // TODO: handle exception
        if (Log.isDebugEnabled()) Log.debug("Error - OrdenAtencionDAOImpl.actualizarEstadoAbandonoTriage");
        Log.error(ex, ex);
        throw ex;
    } finally {
        if (Log.isDebugEnabled()) Log.debug("Final - OrdenAtencionDAOImpl.actualizarEstadoAbandonoTriage");
    }
    return rows==1;
}
```

Fuente: Elaboración propia

En la figura 116 se puede ver el código del modelo de datos para actualizar estado de abandono y su ejecución en la base de datos para luego devolver la respuesta.

Figura 117 Modelo Nivel Porcentual de Pacientes Atendidos en Triage

```
package pe.edu.sanna.triage.model;

public abstract class ItemNivPorcentualReportModel {

    int item;
    String fecha;
    int internista;
    String turno;
    int ctp;
    int cpc;
    double cpca;

    public int getItem() {
        return item;
    }
    public void setItem(int item) {
        this.item = item;
    }
    public String getFecha() {
        return fecha;
    }
    public void setFecha(String fecha) {
        this.fecha = fecha;
    }
    public int getInternista() {
        return internista;
    }
    public void setInternista(int internista) {
        this.internista = internista;
    }
    public String getTurno() {
        return turno;
    }
    public void setTurno(String turno) {
        this.turno = turno;
    }
    public int getCtp() {
        return ctp;
    }
    public void setCtp(int ctp) {
        this.ctp = ctp;
    }
    public int getCpc() {
        return cpc;
    }
    public void setCpc(int cpc) {
        this.cpc = cpc;
    }
    public double getCpca() {
        return cpca;
    }
    public void setCpca(double cpca) {
        this.cpc = cpca;
    }
}
```

Fuente: Elaboración propia

En la figura 117 se puede ver el código del modelo nivel porcentual de pacientes atendidos en triage, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase ItemNivPorcentualReportModel.

Figura 118 Vista Nivel Porcentual de Pacientes Atendido en Triage

```
function LoadDataTableWithAjax(){
    var periodo=document.getElementById('txtTopicos').value;

    // alert(""+periodo);
    $('#data-table-default').dataTable().fnClearTable();
    $('#data-table-default').dataTable().fnDraw();
    $.ajax({
        type: "GET",
        url: "${pageContext.request.contextPath}/ItemNivelPorcentualReport/ListNivelPor/"+ periodo,
        crossDomain: true,
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        async: true,
        success: function (response) {
            //ADD NOTIFICATION
            //ShowMessage(response.message);

            if(response.data.length > 0){
                //UPDATE DATATABLES
                $('#data-table-default').dataTable().fnAddData(response.data);
                //alert(JSON.stringify(response.data));
            }else{
                console.log("No hay Atenciones")
            }
            //UPDATE DATATABLES
        },
        //error: OnErrorCall
    })
}

$("#btnBuscar").click(function() {
    // $('#modal-dialog').modal('show');
    // $('#stack1').modal('show')
    var periodo=document.getElementById('txtTopicos').value;

    if(periodo=="0"){
        $.toast({
            heading: 'Warning',
            text: 'NO HA SELECCIONADO PERIODO',
            position: 'top-center',
            showHideTransition: 'plain',
            icon: 'warning'
        });
    }
});
```

Fuente: Elaboración propia

En la figura 118 se puede ver un fragmento del código de la vista de nivel porcentual de pacientes atendidos en triage el cual se está enviando por Ajax mediante una ruta al controlador los atributos a listar para poder realizar la ejecución con la BD.

Figura 119 Controlador Nivel Porcentual de Pacientes Atendidos en Triage

```
@RequestMapping(value="/ListNivelPor/{periodo}",method=RequestMethod.GET,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> listarNivelPorcentualReporte(@PathVariable String periodo) throws Exception {

    System.out.print("entrooooooooooooooooooooo ");

    Map<String, Object>mapResult=new HashMap<>();

    try {

        List<ItemNivPorcentualReporteBean> lstItemRepot =itemNivelPorcentualReportService.listarNivelPorcentualReporte(periodo);

        mapResult.put(Constants.STATUS_TEXT, true);
        mapResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_READ.replace("{0}", String.valueOf(lstItemRepot.size())));
        mapResult.put("data", lstItemRepot);

        System.out.print("oscasito: "+lstItemRepot.size());

    } catch (Exception e) {
        Log.error("ItemNivPorcentualReportController.listarNivelPorcentualReporte - Error");
        Log.error(e.getMessage());
        mapResult.put(Constants.STATUS_TEXT, false);
        mapResult.put(Constants.MESSAGE_TEXT, e.getLocalizedMessage());
        // TODO: handle exception
    }
    return mapResult;
}
```

Fuente: Elaboración propia

En la figura 119 se puede ver el código del controlador de nivel porcentual de pacientes atendidos en triage para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de listar de nivel porcentual de pacientes atendidos en triage o mensaje de error.

Figura 120 Ejecución BD Nivel Porcentual Pacientes Atendidos en Triage

```
@Override
public List<ItemNivPorcentualReporteBean> listarNivelPorcentualReporte(String periodo) throws Exception {
    // TODO Auto-generated method stub
    if (Log.isDebugEnabled()) Log.debug("Inicio - ItemNivelPorcentualReportDAOImpl.listarNivelPorcentualReporte");
    List<ItemNivPorcentualReporteBean> lstOrden;
    String SQL = "EXEC SP_LISTAR_NIVEL_PORCENTUAL ?";
    try {

        lstOrden = jdbcTemplate.query(SQL, new Object[] { periodo}, new ResultSetExtractor<List<ItemNivPorcentualReporteBean>>() {
            @Override
            public List<ItemNivPorcentualReporteBean> extractData(ResultSet rs) throws SQLException, DataAccessException {
                // TODO Auto-generated method stub
                List<ItemNivPorcentualReporteBean> lista = new ArrayList<>();
                while (rs.next()) {
                    ItemNivPorcentualReporteBean orden= new ItemNivPorcentualReporteBean();

                    orden.setItem(rs.getInt("item"));
                    orden.setFecha(rs.getString("fecha"));
                    orden.setInternista(rs.getInt("internista"));
                    orden.setTurno(rs.getString("turno"));
                    orden.setCtp(rs.getInt("ctp"));
                    orden.setCpc(rs.getInt("cpc"));
                    orden.setCpca(rs.getDouble("cpca"));

                    lista.add(orden);
                }
                return lista;
            }
        });

    } catch (Exception ex) {
        // TODO: handle exception
        if (Log.isDebugEnabled()) Log.debug("Error - ItemNivelPorcentualReportDAOImpl.listarNivelPorcentualReporte");
        Log.error(ex, ex);
        throw ex;
    } finally {
        if (Log.isDebugEnabled()) Log.debug("Final - ItemNivelPorcentualReportDAOImpl.listarNivelPorcentualReporte");
    }
    return lstOrden;
}
```

Fuente: Elaboración propia

En la figura 120 se puede ver el código del modelo de datos para listar nivel porcentual de pacientes atendidos en triage y su ejecución en la base de datos para luego devolver la respuesta.

Figura 121 Modelo Reporte Triage

```
package pe.edu.sanna.triage.model;

public class ItemReporteGeneralTriage {
    int periodo;
    int ordenatencion;
    int historiaclinica;
    String seguro;
    String nombrecompleto;
    String documento;
    String sexo;
    int edad;
    String estadotriage;
    String motivo;
    String detalllediscriminador;
    String alergias;
    String internista;
    String prioridad;
    String topicoderivar;
    String fechaaperturaatriage;
    String fecharegistroatriage;

    public String getInternista() {
        return internista;
    }
    public void setInternista(String internista) {
        this.internista = internista;
    }
    public int getPeriodo() {
        return periodo;
    }
    public void setPeriodo(int periodo) {
        this.periodo = periodo;
    }
    public int getOrdenatencion() {
        return ordenatencion;
    }
    public void setOrdenatencion(int ordenatencion) {
        this.ordenatencion = ordenatencion;
    }
    public int getHistoriaclinica() {
        return historiaclinica;
    }
    public void setHistoriaclinica(int historiaclinica) {
        this.historiaclinica = historiaclinica;
    }
    public String getSeguro() {
        return seguro;
    }
    public void setSeguro(String seguro) {
        this.seguro = seguro;
    }
    public String getNombrecompleto() {
        return nombrecompleto;
    }
}
```

Fuente: Elaboración propia

En la figura 121 se puede ver el código del modelo reporte triage, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase ItemReportGeneralTriageModel.

Figura 122 Vista Reporte Triage

```
function LoadDataTableWithAjax(){
    var periodo=document.getElementById('txtTemas').value;
    // alert(""+periodo);
    $('#data-table-default').dataTable().fnClearTable();
    $('#data-table-default').dataTable().fnDraw();
    $.ajax({
        type: "GET",
        url: "${pageContext.request.contextPath}/ItemReporteGeneralTriage/ListReportTriage/" + periodo,
        crossDomain: true,
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        async: true,
        success: function (response) {
            //ADD NOTIFICACION
            //ShowMessage(response.message);

            if(response.data.length > 0){
                //var valor=JSON.stringify(response);
                //alert(""+valor[1].horadiff);
                //UPDATE DATATABLES
                var datito = response.data;
                //alert("");
                $('#data-table-default').dataTable().fnAddData(response.data);
            }else{
                console.log("No hay data")
            }
        }
    });
}

$( "#btnBuscar" ).click(function() {
    // $('#modal-dialog').modal('show');
    // $('#stack1').modal('show')
    var periodo=document.getElementById('txtTemas').value;

    if(periodo=="0"){
        $.toast({
            heading: 'Warning',
            text: 'NO HA SELECCIONADO PERIODO',
            position: 'top-center',
            showHideTransition: 'plain',
            icon: 'warning'
        });
    }
    LoadDataTableWithAjax()
});
```

Fuente: Elaboración propia

En la figura 122 se puede ver un fragmento del código de la vista de reporte triage el cual se está enviando por Ajax mediante una ruta al controlador los atributos a listar para poder realizar la ejecución con la BD.

Figura 123 Controlador Reporte Triage

```
@RequestMapping(value="/listReportTriage/{periodo}",method=RequestMethod.GET,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept-application/json")
public @ResponseBody Map<String, Object> listarReporteGeneralTriage(@PathVariable String periodo) throws Exception {

    Map<String, Object> mapResult=new HashMap<>();

    try {

        List<ItemReporteGeneralTriageBean> lstItemReport = itemReporteGeneralTriageService.listarReporteGeneralTriage(periodo);

        mapResult.put(Constantes.STATUS_TEXT, true);
        mapResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_READ.replace("{0}", String.valueOf(lstItemReport.size())));
        mapResult.put("data", lstItemReport);

    } catch (Exception e) {
        Log.error("ItemReporteGeneralTriageController.listarReporteGeneralTriage - Error");
        Log.error(e.getMessage());
        mapResult.put(Constantes.STATUS_TEXT, false);
        mapResult.put(Constantes.MESSAGE_TEXT, e.getLocalizedMessage());
        // TODO: handle exception
    }
    return mapResult;
}
```

Fuente: Elaboración propia

En la figura 123 se puede ver el código del controlador reporte triage para poder realizar él envió a la BD para la validación de datos y obtener la respuesta de listar reporte triage o mensaje de error.

Figura 124 Ejecución BD Reporte Triage

```
@Override
public List<ItemReporteGeneralTriageBean> listarReporteGeneralTriage(String periodo) throws Exception {
    if (Log.isDebugEnabled())
        Log.debug("Inicio - ItemReporteGeneralTriageDAOImpl.listarReporteGeneralTriage");
    List<ItemReporteGeneralTriageBean> lstOrden;
    String SQL ="EXEC SP_LISTAR_REPORTES_TRIAGE ? ";
    try {
        lstOrden=jdbcTemplate.query(SQL, new Object[] {periodo}, new ResultSetExtractor<List<ItemReporteGeneralTriageBean>>() {
            @Override
            public List<ItemReporteGeneralTriageBean> extractData(ResultSet rs) throws SQLException, DataAccessException {
                List<ItemReporteGeneralTriageBean> lista = new ArrayList<>();
                while (rs.next()) {
                    ItemReporteGeneralTriageBean orden= new ItemReporteGeneralTriageBean();

                    orden.setHistoriaclinica(rs.getInt("historiaclinica"));
                    orden.setSeguro(rs.getString("seguro"));
                    orden.setNombrecompleto(rs.getString("nombrecompleto"));
                    orden.setDocumento(rs.getString("documento"));
                    orden.setSexo(rs.getString("sexo"));
                    orden.setEdad(rs.getInt("edad"));
                    orden.setEstadotriage(rs.getString("estadotriage"));
                    orden.setMotivo(rs.getString("motivo"));
                    orden.setDetalladiscriminador(rs.getString("detailediscriminador"));
                    orden.setInternista(rs.getString("internista"));
                    orden.setPrioridad(rs.getString("prioridad"));
                    orden.setTopicoderivar(rs.getString("topicoderivar"));
                    orden.setFechaaperturatriage(rs.getString("fechaaperturatriage"));
                    orden.setFecharegistrotriage(rs.getString("fecharegistrotriage"));
                    lista.add(orden);
                }
                return lista;
            }
        });
    } catch (Exception ex) {
        // TODO: handle exception
        if (Log.isDebugEnabled())
            Log.debug("Error - ItemReporteGeneralTriageDAOImpl.listarReporteGeneralTriage");
        Log.error(ex, ex);
        throw ex;
    } finally {
        if (Log.isDebugEnabled()) Log.debug("Final - ItemReporteGeneralTriageDAOImpl.listarReporteGeneralTriage");
    }
}
```

Fuente: Elaboración propia

En la figura 124 se puede ver el código del modelo de datos para listar reporte triage y su ejecución en la base de datos para luego devolver la respuesta.

Figura 125 Modelo Enfermedad Actual

```
package pe.edu.sanna.triage.model;

public abstract class EnfermedadActualModel {

    int emf_idordenatencion;
    String emf_motivoingreso;
    String emf_formainicio;
    String emf_cursoenfermedad;
    int emf_tiempoenfermedad;
    String emf_unidadenfermedad;
    String emf_relato cronologico;
    String emf_examen clinico;
    String emf_usuarioCreacion;

    public int getEmf_idordenatencion() {
        return emf_idordenatencion;
    }
    public void setEmf_idordenatencion(int emf_idordenatencion) {
        this.emf_idordenatencion = emf_idordenatencion;
    }
    public String getEmf_motivoingreso() {
        return emf_motivoingreso;
    }
    public void setEmf_motivoingreso(String emf_motivoingreso) {
        this.emf_motivoingreso = emf_motivoingreso;
    }
    public String getEmf_formainicio() {
        return emf_formainicio;
    }
    public void setEmf_formainicio(String emf_formainicio) {
        this.emf_formainicio = emf_formainicio;
    }
    public String getEmf_cursoenfermedad() {
        return emf_cursoenfermedad;
    }
    public void setEmf_cursoenfermedad(String emf_cursoenfermedad) {
        this.emf_cursoenfermedad = emf_cursoenfermedad;
    }
    public int getEmf_tiempoenfermedad() {
        return emf_tiempoenfermedad;
    }
    public void setEmf_tiempoenfermedad(int emf_tiempoenfermedad) {
        this.emf_tiempoenfermedad = emf_tiempoenfermedad;
    }
    public String getEmf_unidadenfermedad() {
        return emf_unidadenfermedad;
    }
    public void setEmf_unidadenfermedad(String emf_unidadenfermedad) {
        this.emf_unidadenfermedad = emf_unidadenfermedad;
    }
    public String getEmf_relato cronologico() {
        return emf_relato cronologico;
    }
}
```

Fuente: Elaboración propia

En la figura 125 se puede ver el código del modelo enfermedad actual, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase EnfermedadActualModel.

Figura 126 Vista Enfermedad Actual

```
$.ajax({
    type: (op.length == 0 ? "POST" : "PUT"), //SI TIENE ES UNA ACTUALIZACION
    url: "${pageContext.request.contextPath}/EnfermedadActual/RegisUpd",
    crossDomain: true,
    data: formToJson,
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    success: function (response) {

        if(response.status == true){

            $.toast({
                heading: 'Exito',
                text: 'ENFERMEDAD ACTUAL REGISTRADO SATISFACOTRIAMENTE',
                position: 'top-center',
                icon: 'success'
            });

            LimpiarTxtEnferActual();
            $("#formMantEnfermedadAct")[0].reset();

            $("#Contenfermedad").load('../EnfermedadActual/WebEmergencia');

        }else{

            $.toast({
                heading: 'Error',
                text: 'NO SE PUDO REGISTRAR',
                position: 'top-center',
                icon: 'error'
            });

        }

    },
});
```

Fuente: Elaboración propia

En la figura 126 se puede ver un fragmento del código de la vista de gestionar enfermedad actual el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 127 Controlador Enfermedad Actual

```
@RequestMapping(value="/Registrar",method=RequestMethod.POST,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept-application/json")
@ResponseBody Map<String, Object> getRegistrarEnfermedadActual(HttpServletRequest request, HttpServletResponse response) throws Exception {
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        String line;
        StringBuilders jsonEnviado = new StringBuilders();
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        EnfermedadActualBean enfermedadActualBean = (EnfermedadActualBean) new JsonSerializer().deserialize(jsonEnviado.toString(), EnfermedadActualBean.class);
        result = enfermedadActualService.registrarEnfermedadActual(enfermedadActualBean);
        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}

// METODO PUT ACTUALIZAR UN REGISTRO <- JSON
@RequestMapping(value="/Registrar",method=RequestMethod.PUT, produces = MediaType.APPLICATION_JSON_VALUE, headers = "Accept-application/json")
@ResponseBody Map<String, Object> ActualizarEnfermedadActual(HttpServletRequest request, HttpServletResponse response) throws Exception {
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        String line;
        StringBuilders jsonEnviado = new StringBuilders();
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        EnfermedadActualBean UpEnferme = (EnfermedadActualBean) new JsonSerializer().deserialize(jsonEnviado.toString(), EnfermedadActualBean.class);
        // ACTUALIZAR USUARIO
        result = enfermedadActualService.actualizarEnfermedadActual(UpEnferme);
        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_UPDATE.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 127 se puede ver el código del controlador enfermedad actual para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de registro de enfermedad actual o mensaje de error.

Figura 128 Ejecución BD Registrar Enfermedad Actual

```
@Override
public boolean insertarEnfermedadActual(EnfermedadActualBean enfermedadActualBean) throws Exception {
    // TODO Auto-generated method stub
    if(Log.isDebugEnabled()) Log.debug("Inicio - EnfermedadActualDAOImpl.insertarEnfermedadActual");
    String SQL = "EXEC SP_TR_INSERT_ENFERMEDAD_ACTUAL ?,?,?,?,?,?,?,?";
    try {
        rows= jdbcTemplate.update(SQL.toString(),new Object[] {
            enfermedadActualBean.getIdordenatencion(),
            enfermedadActualBean.getIdmotivoingreso(),
            enfermedadActualBean.getIdformainicio(),
            enfermedadActualBean.getIdcursoenfermedad(),
            enfermedadActualBean.getIdtiempoenfermedad(),
            enfermedadActualBean.getIdunidadesenfermedad(),
            enfermedadActualBean.getIdrelatocronologico(),
            enfermedadActualBean.getIdexamenclinico(),
            enfermedadActualBean.getIdusuarioCreacion()
        });
    } catch (Exception ex) {
        if(Log.isDebugEnabled()) Log.debug("Inicio - EnfermedadActualDAOImpl.insertarEnfermedadActual");
        Log.error(ex,ex);
        throw ex;
    }
    // TODO: handle exception
    finally {
        if(Log.isDebugEnabled()) Log.debug("Inicio - EnfermedadActualDAOImpl.insertarEnfermedadActual");
    }
    return rows==1;
}
```

Fuente: Elaboración propia

En la figura 128 se puede ver el código del modelo de datos para registrar enfermedad actual y su ejecución en la base de datos para luego devolver la respuesta.

Figura 129 Modelo Funciones Vitales

```
package pe.edu.sanna.triage.model;

public abstract class FuncionesVitalesModel {

    int fv_idordenatencion;
    int fv_pa;

    int fv_pamhg;
    int fv_fr;
    double fv_temperatura;
    int fv_sabo2;
    double fv_peso;
    double fv_talla;
    double fv_imc;
    String fv_usuarioCreacion;
    int fv_fc;

    public int getFv_fc() {
        return fv_fc;
    }
    public void setFv_fc(int fv_fc) {
        this.fv_fc = fv_fc;
    }
    public int getFv_idordenatencion() {
        return fv_idordenatencion;
    }
    public void setFv_idordenatencion(int fv_idordenatencion) {
        this.fv_idordenatencion = fv_idordenatencion;
    }
    public int getFv_pa() {
        return fv_pa;
    }
    public void setFv_pa(int fv_pa) {
        this.fv_pa = fv_pa;
    }
    public int getFv_pamhg() {
        return fv_pamhg;
    }
    public void setFv_pamhg(int fv_pamhg) {
        this.fv_pamhg = fv_pamhg;
    }
    public int getFv_fr() {
        return fv_fr;
    }
    public void setFv_fr(int fv_fr) {
        this.fv_fr = fv_fr;
    }
    public double getFv_temperatura() {
        return fv_temperatura;
    }
    public void setFv_temperatura(double fv_temperatura) {
        this.fv_temperatura = fv_temperatura;
    }
}
```

Fuente: Elaboración propia

En la figura 129 se puede ver el código del modelo funciones vitales, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase FuncionesVitalesModel.

Figura 130 Vista Funciones Vitales

```
$.ajax({
    type: (op.length == 0 ? "POST" : "PUT"), //SI TIENE ES UNA ACTUALIZACION
    url: "${pageContext.request.contextPath}/FuncionesVitales/RegisUpd",
    crossDomain: true,
    data: form103JSON,
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    success: function (response) {

        if(response.status === true){

            $.toast({
                heading: 'Exito',
                text: 'FUNCIONES VITALES REGISTRADO SATISFACTORIAMENTE',
                position: 'top-center',
                icon: 'success'
            });

            LimpiarTxtFuncionesVitales();
            $("#formMantFunVitales")[0].reset();

            $("#ContenFuncVitales").load('../FuncionesVitales/WebEmergencia');

        }else{

            $.toast({
                heading: 'Error',
                text: 'NO SE PUUDO REGISTRAR',
                position: 'top-center',
                icon: 'error'
            });

        }

    },
});
```

Fuente: Elaboración propia

En la figura 130 se puede ver un fragmento del código de la vista de gestionar funciones vitales el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 131 Controlador Funciones Vitales

```
@RequestMapping(value="/RegisUpd",method=RequestMethod.POST,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept-application/json")
public @ResponseBody Map<String, Object> getRegistrarEnfermedadActual(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append("\n");
        }

        FuncionesVitalesBean funcionesVitalesBean = (FuncionesVitalesBean) new JsonSerializer().deserialize(jsonEnviado.toString(),FuncionesVitalesBean.class);
        result = funcionesVitalesService.registrarFuncionesVitales(funcionesVitalesBean);

        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));

    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}

@RequestMapping(value="/RegisUpd",method=RequestMethod.PUT,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept-application/json")
public @ResponseBody Map<String, Object> getActualizarEnfermedadActual(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append("\n");
        }

        FuncionesVitalesBean funcionesVitalesBean = (FuncionesVitalesBean) new JsonSerializer().deserialize(jsonEnviado.toString(),FuncionesVitalesBean.class);
        result = funcionesVitalesService.actualizarFuncionesVitales(funcionesVitalesBean);

        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));

    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 131 se puede ver el código del controlador funciones vitales para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de registro de funciones vitales o mensaje de error.

Figura 132 Ejecución BD Registrar Funciones Vitales

```
@Override
public boolean InsertarFuncionesVitales(FuncionesVitalesBean funcionesVitalesBean) throws Exception {
    // TODO Auto-generated method stub
    if(Log.isDebugEnabled()) Log.debug("Inicio - FuncionesVitalesDAOImpl.InsertarFuncionesVitales");
    String SQL = "EXEC SP_TR_INSERT_FUNCIONES_VITALES ?,?,?,?,?,?,?,?,?,?";

    try {
        rows= jdbcTemplate.update(SQL.toString(),new Object[] {

            funcionesVitalesBean.getFv_idordenatencion(),
            funcionesVitalesBean.getFv_pa(),
            funcionesVitalesBean.getFv_pamhg(),
            funcionesVitalesBean.getFv_fr(),
            funcionesVitalesBean.getFv_temperatura(),
            funcionesVitalesBean.getFv_sato2(),
            funcionesVitalesBean.getFv_peso(),
            funcionesVitalesBean.getFv_talla(),
            funcionesVitalesBean.getFv_imc(),
            funcionesVitalesBean.getFv_usuarioCreacion(),
            funcionesVitalesBean.getFv_fc()

        });

    } catch (Exception ex) {
        if(Log.isDebugEnabled()) Log.debug("Inicio - FuncionesVitalesDAOImpl.InsertarFuncionesVitales");
        Log.error(ex,ex);
        throw ex;
    }
    // TODO: handle exception
    finally {
        if(Log.isDebugEnabled()) Log.debug("Inicio - FuncionesVitalesDAOImpl.InsertarFuncionesVitales");
    }
    return rows==1;
}
```

Fuente: Elaboración propia

En la figura 132 se puede ver el código del modelo de datos para registrar funciones vitales y su ejecución en la base de datos para luego devolver la respuesta.

Figura 133 Modelo Diagnostico

```
package pe.edu.sanna.triage.model;

public abstract class CatalogoDiagnosticoModel {

    int catDia_IdOrdenAten;

    int catDia_Iddiagnostico;
    String catDia_CodigoPadre;
    String catDia_Descripcion;
    String catDia_DetDiagnostico;
    String catDia_GradoAfeccion;
    String catDia_Principal;
    String catDia_Accion;

    String catDia_UsuarioCra;
    int catposicion;

    public int getCatposicion() {
        return catposicion;
    }
    public void setCatposicion(int catposicion) {
        this.catposicion = catposicion;
    }
    public int getCatDia_IdOrdenAten() {
        return catDia_IdOrdenAten;
    }
    public void setCatDia_IdOrdenAten(int catDia_IdOrdenAten) {
        this.catDia_IdOrdenAten = catDia_IdOrdenAten;
    }
    public String getCatDia_DetDiagnostico() {
        return catDia_DetDiagnostico;
    }
    public void setCatDia_DetDiagnostico(String catDia_DetDiagnostico) {
        this.catDia_DetDiagnostico = catDia_DetDiagnostico;
    }
    public String getCatDia_GradoAfeccion() {
        return catDia_GradoAfeccion;
    }
    public void setCatDia_GradoAfeccion(String catDia_GradoAfeccion) {
        this.catDia_GradoAfeccion = catDia_GradoAfeccion;
    }
    public String getCatDia_Principal() {
        return catDia_Principal;
    }
    public void setCatDia_Principal(String catDia_Principal) {
        this.catDia_Principal = catDia_Principal;
    }
    public String getCatDia_Accion() {
        return catDia_Accion;
    }
    public void setCatDia_Accion(String catDia_Accion) {
        this.catDia_Accion = catDia_Accion;
    }
}
```

Fuente: Elaboración propia

En la figura 133 se puede ver el código del modelo diagnóstico, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase DiagnosticoModel.

Figura 134 Vista Diagnóstico

```
$.ajax({
    type: "POST",
    url: "${pageContext.request.contextPath}/CatalogoDiagnostico/Guardar",
    crossDomain: true,
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    data: JSON.stringify(gc),
    async: false,
    success: function (response) {

        if(!response.status){

            alert("error");
            return false;
        }
        //CARGAR DATATABLE
        $.toast({
            heading: 'Exito',
            text: 'DIAGNOSTICO REGISTRADO SATISFACTORIAMENTE',
            position: 'bottom-right',
            icon: 'success'
        });

        LoadDataTableAjaxPrincipal();
        //window.location.reload();
        //$("#Contenddiag").load('../CatalogoDiagnostico/WebEmergencia');

    },
    error: OnErrorCall
});
});
```

Fuente: Elaboración propia

En la figura 134 se puede ver un fragmento del código de la vista de gestionar diagnóstico el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 135 Controlador Diagnóstico

```
// METODO POST INSERTAR UN REGISTRO <- JSON
@RequestMapping(value="/Guardar", method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_VALUE, headers = "Accept-application/json")
public @ResponseBody Map<String, Object> postDiagnostico(HttpServletRequest request, HttpServletResponse response) throws Exception {
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY JSON
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream()));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        //SERIALIZAMOS EL JSON -> CLASS
        CatalogoDiagnosticoBean listDetGridDiag = (CatalogoDiagnosticoBean) new JsonSerializer().deserialize(jsonEnviado.toString(), CatalogoDiagnosticoBean.class);
        //SERVICE INSERTAR USUARIO
        result = catalogoDiagnosticoService.insertarDiagnostico(listDetGridDiag);
        //mapaResult.put(Constants.STATUS_TEXT, result);
        //mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? listDetGridDiag.getGic_correl(): "0")));
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));

    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 135 se puede ver el código del controlador diagnóstico para poder realizar el envío a la BD para la validación de datos y obtener la respuesta de registro de diagnóstico o mensaje de error.

Figura 136 Ejecución BD Registrar Diagnóstico

```
@Override
public boolean insDiagnosticoDetalle(CatalogoDiagnosticoBean catalogoDiagnosticoBean) throws Exception {

    if (log.isDebugEnabled()) log.debug("Inicio - CatalogoDiagnosticoDAOImpl.insDiagnosticoDetalle");
    String SQL = "EXEC SP_INSERT_DIAGNOSTICO_CIE10 ?,?,?,?,?,?" ;
    try {
        rows = jdbcTemplate.update(SQL.toString(), new Object[] {
            catalogoDiagnosticoBean.getCatDia_IdOrdenAten(),
            catalogoDiagnosticoBean.getCatDia_Iddiagnostico(),
            catalogoDiagnosticoBean.getCatDia_DetDiagnostico(),
            catalogoDiagnosticoBean.getCatDia_GradoAfeccion(),
            catalogoDiagnosticoBean.getCatDia_Principal(),
            catalogoDiagnosticoBean.getCatDia_UsuarioCra()
        });
    } catch (Exception ex) {
        if (log.isDebugEnabled()) log.debug("Error - CatalogoDiagnosticoDAOImpl.insDiagnosticoDetalle");
        log.error(ex, ex);
        throw ex;
    } finally {
        if (log.isDebugEnabled()) log.debug("Final - CatalogoDiagnosticoDAOImpl.insDiagnosticoDetalle");
    }
    return rows > 0;
}
```

Fuente: Elaboración propia

En la figura 136 se puede ver el código del modelo de datos para registrar diagnóstico y su ejecución en la base de datos para luego devolver la respuesta.

Figura 137 Modelo Examen de Apoyo al Diagnostico

```
package pe.edu.sanna.triage.model;

public abstract class ExamenApoyoModel {

    int codigoSegus_Examen;
    String nom_Examen;
    String servi_Examen;
    int idorden_Exa;
    int cantidad;
    String fechsoli_Exa;
    String espec_Exa;
    String usuaCreacion_Exa;

    public int getCodigoSegus_Examen() {
        return codigoSegus_Examen;
    }
    public void setCodigoSegus_Examen(int codigoSegus_Examen) {
        this.codigoSegus_Examen = codigoSegus_Examen;
    }
    public String getNom_Examen() {
        return nom_Examen;
    }
    public void setNom_Examen(String nom_Examen) {
        this.nom_Examen = nom_Examen;
    }
    public String getServu_Examen() {
        return servi_Examen;
    }
    public void setServu_Examen(String servi_Examen) {
        this.servi_Examen = servi_Examen;
    }
    public int getIdorden_Exa() {
        return idorden_Exa;
    }
    public void setIdorden_Exa(int idorden_Exa) {
        this.idorden_Exa = idorden_Exa;
    }
    public int getCantidad() {
        return cantidad;
    }
    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }
    public String getFechsoli_Exa() {
        return fechsoli_Exa;
    }
    public void setFechsoli_Exa(String fechsoli_Exa) {
        this.fechsoli_Exa = fechsoli_Exa;
    }
    public String getEspec_Exa() {
        return espec_Exa;
    }
    public void setEspec_Exa(String espec_Exa) {
        this.espec_Exa = espec_Exa;
    }
}
```

Fuente: Elaboración propia

En la figura 137 se puede ver el código del modelo exámenes de apoyo al diagnóstico, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase ExamenApoyoModel.

Figura 138 Vista Examen de Apoyo al Diagnostico

```
//gc.catDia_UsuarioCra=UsuCreacion;
//alert(JSON.stringify(gc));

$.ajax({
    type: "POST",
    url: "${pageContext.request.contextPath}/ExamenApoyo/Guardar",
    crossDomain: true,
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    data: JSON.stringify(gc),
    async: false,
    success: function (response) {
        if(!response.status){
            return false;
        }
        //CARGAR DATATABLE
        $.toast({
            heading: 'Exito',
            text: 'EXAMEN DE APOYO AL DIAGNOSTICO REGISTRADO SATISFACTORIAMENTE',
            position: 'bottom-right',
            icon: 'success'
        });
        LoadDataTableAjaxPrincipal();
    },
    // error: OnErrorCall
});
});
```

Fuente: Elaboración propia

En la figura 138 se puede ver un fragmento del código de la vista de gestionar exámenes de apoyo al diagnóstico el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 139 Controlador Examen de Apoyo al Diagnóstico

```
// METODO POST INSERTAR UN REGISTRO <- JSON
@RequestMapping(value="/Guardar", method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_VALUE, headers = "Accept-application/json")
public @ResponseBody Map<String, Object> postDiagnostico(HttpServletRequest request, HttpServletResponse response) throws Exception {
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        // LECTURA DEL REQUEST BODY JSON
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream()));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        // SERIALIZAMOS EL JSON -> CLASS
        ExamenesApoyoBean listDetExamenApoyo = (ExamenesApoyoBean) new JsonSerializer().deserialize(jsonEnviado.toString(), ExamenesApoyoBean.class);
        // SERVICE INSERTAR USUARIO
        result = examenesApoyoService.insertarExamenApoyo(listDetExamenApoyo);
        mapaResult.put(Constants.STATUS_TEXT, result);
        // mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? listDetGridDiag.getGic_correl(): "0")));
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 139 se puede ver el código del controlador exámenes de apoyo al diagnóstico para poder realizar el envío a la BD para la validación de datos y obtener la respuesta de registro de exámenes de apoyo al diagnóstico o mensaje de error.

Figura 140 Ejecución Registrar Examen de Apoyo al Diagnóstico

```
@Override
public boolean insExamenApoyoDetalle(ExamenesApoyoBean examenesApoyoBean) throws Exception {
    if (Log.isDebugEnabled()) Log.debug("Inicio - ExamenesApoyoDAOImpl.insExamenApoyoDetalle");
    String SQL = "EXEC SP_INSERT_EXAMENES_APOYO ?,?,?,?";
    try {
        rows = jdbcTemplate.update(SQL.toString(), new Object[] {
            examenesApoyoBean.getCodigoSegus_Examen(),
            examenesApoyoBean.getIdorden_Exa(),
            examenesApoyoBean.getCantidad(),
            examenesApoyoBean.getEspec_Exa(),
            examenesApoyoBean.getUsuaCreacion_Exa()
        });
    } catch (Exception ex) {
        if (Log.isDebugEnabled()) Log.debug("Error - ExamenesApoyoDAOImpl.insExamenApoyoDetalle");
        Log.error(ex, ex);
        throw ex;
    } finally {
        if (Log.isDebugEnabled()) Log.debug("Final - ExamenesApoyoDAOImpl.insExamenApoyoDetalle");
    }
    return rows > 0;
}
```

Fuente: Elaboración propia

En la figura 140 se puede ver el código del modelo de datos para registrar exámenes de apoyo al diagnóstico y su ejecución en la base de datos para luego devolver la respuesta.

Figura 141 Modelo Receta

```
package pe.edu.sanna.triage.modelo;

public class SuministrarRecetaModel {

    int idComponente ;
    String nombreComponente;
    int idOrden_Sumi;
    String unidad_Sumi;
    String via_Sumi;
    int tratamiento_Sumi;
    String u_trata_Sumi;
    int catat_Sumi;
    String u_cada_Sumi;
    int cantidad_SumiReceta;

    int dias_SumiReceta;
    String indica_especifica_Sumi;
    String especi_SumiReceta;
    String usuSumiReceta;

    public String getUnidad_Sumi() {
        return unidad_Sumi;
    }
    public void setUnidad_Sumi(String unidad_Sumi) {
        this.unidad_Sumi = unidad_Sumi;
    }
    public String getVia_Sumi() {
        return via_Sumi;
    }
    public void setVia_Sumi(String via_Sumi) {
        this.via_Sumi = via_Sumi;
    }
    public int getTratamiento_Sumi() {
        return tratamiento_Sumi;
    }
    public void setTratamiento_Sumi(int tratamiento_Sumi) {
        this.tratamiento_Sumi = tratamiento_Sumi;
    }
    public String getU_trata_Sumi() {
        return u_trata_Sumi;
    }
    public void setU_trata_Sumi(String u_trata_Sumi) {
        this.u_trata_Sumi = u_trata_Sumi;
    }
    public int getCatat_Sumi() {
        return catat_Sumi;
    }
    public void setCatat_Sumi(int catat_Sumi) {
        this.catat_Sumi = catat_Sumi;
    }
    public String getU_cada_Sumi() {
        return u_cada_Sumi;
    }
}
```

Fuente: Elaboración propia

En la figura 141 se puede ver el código del modelo receta, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase SuministrarRecetaModel.

Figura 142 Vista Receta

```
$( 'body' ).on( 'click', '#btnRegistrarSumiReceta', function (e) {
    var gc = new Object();
    gc.gic_detalle= dtSuministrarRecePri.rows().data().toArray();
    //alert(JSON.stringify(gc));
    $.ajax({
        type: "POST",
        url: "${pageContext.request.contextPath}/SuministrarReceta/Guardar",
        crossDomain: true,
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        data : JSON.stringify(gc),
        async: false,
        success: function (response) {

            if(!response.status){

                alert("error");
                return false;
            }
            //CARGAR DATATABLE
            $.toast({
                heading: 'Exito',
                text: 'RECETA REGISTRADO SATISFACTORIAMENTE',
                position: 'bottom-right',
                icon: 'success'
            });

            LoadDataFableAjaxPrincipal()

        },
        // error: OnErrorCall
    });
});
```

Fuente: Elaboración propia

En la figura 142 se puede ver un fragmento del código de la vista de gestionar receta el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 143 Controlador Receta

```
// METODO POST INSERTAR UN REGISTRO <- JSON
@RequestMapping(value="/guardar", method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_VALUE, headers = "Accept-application/json")
public @ResponseBody Map<String, Object> postSumistrarReceta(HttpServletRequest request, HttpServletResponse response) throws Exception {
    Map<String, Object> mapaResult = new HashMap<>();
    try {

        //LECTURA DEL REQUEST BODY JSON
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(), "UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        //SERIALIZAMOS EL JSON -> CLASS
        SuministrarRecetaBean listDetSuminisReceta = (SuministrarRecetaBean) new JsonSerializer().deserialize(jsonEnviado.toString(), SuministrarRecetaBean.class);
        //SERVICE INSERTAR USUARIO
        result = suministrarRecetaService.insertarSumistraReceta(listDetSuminisReceta);
        //-----
        mapaResult.put(Constantes.STATUS_TEXT, result);
        //mapaResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? listDetGridDiag.getGic_correl(): "0")));
        mapaResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));

    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constantes.STATUS_TEXT, false);
        mapaResult.put(Constantes.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 143 se puede ver el código del controlador receta para poder realizar el envío a la BD para la validación de datos y obtener la respuesta de registro de receta o mensaje de error.

Figura 144 Ejecución BD Registrar Receta

```
@Override
public boolean insSumistrarRecetaDetalle(SuministrarRecetaBean suministrarRecetaBean) throws Exception {
    if (Log.isDebugEnabled()) Log.debug("Inicio - SumistrarRecetaDAOImpl.insSumistrarRecetaDetalle");
    String SQL = "EXEC SP_INSERT_SUMISTRAR_RECETA ?,?,?,?,?,?,?,?,?,?,?";
    try {
        rows = jdbcTemplate.update(SQL.toString(), new Object[] {
            suministrarRecetaBean.getldOrden_Sumi(),
            suministrarRecetaBean.getldComponente(),
            suministrarRecetaBean.getUnidad_Sumi(),
            suministrarRecetaBean.getVia_Sumi(),
            suministrarRecetaBean.getTratamiento_Sumi(),
            suministrarRecetaBean.getU_trata_Sumi(),
            suministrarRecetaBean.getCatata_Sumi(),
            suministrarRecetaBean.getU_cada_Sumi(),
            suministrarRecetaBean.getCantidad_SumiReceta(),
            suministrarRecetaBean.getDias_SumiReceta(),
            suministrarRecetaBean.getIndica_espepcifica_Sumi(),
            suministrarRecetaBean.getEspeci_SumiReceta(),
            suministrarRecetaBean.getUsuSumiReceta()
        });
    } catch (Exception ex) {
        if (Log.isDebugEnabled()) Log.debug("Error - SumistrarRecetaDAOImpl.insSumistrarRecetaDetalle");
        Log.error(ex, ex);
        throw ex;
    } finally {
        if (Log.isDebugEnabled()) Log.debug("Final - SumistrarRecetaDAOImpl.insSumistrarRecetaDetalle");
    }
    return rows == 1;
}
```

Fuente: Elaboración propia

En la figura 144 se puede ver el código del modelo de datos para registrar receta y su ejecución en la base de datos para luego devolver la respuesta.

Figura 145 Modelo Descanso Medico

```
package pe.edu.sanna.triage.modelo;
import com.fasterxml.jackson.annotation.JsonFormat;
public abstract class DescansoMedicoModel {

    @JsonFormat(shape = JsonFormat.Shape.STRING, pattern="yyyy-MM-dd", timezone = "GMT-5")

    int des_idordenatencion;
    String des_fechaactual;
    String des_observacion;
    String des_f_del;
    String des_f_al;
    int des_diasdescanso;
    String nombrecompleto;
    int edad;
    String sexo;
    public String getNombrecompleto() {
        return nombrecompleto;
    }
    public void setNombrecompleto(String nombrecompleto) {
        this.nombrecompleto = nombrecompleto;
    }
    public int getEdad() {
        return edad;
    }
    public void setEdad(int edad) {
        this.edad = edad;
    }
    public String getSexo() {
        return sexo;
    }
    public void setSexo(String sexo) {
        this.sexo = sexo;
    }
    String des_usuarioCreacion;
    public int getDes_idordenatencion() {
        return des_idordenatencion;
    }
    public void setDes_idordenatencion(int des_idordenatencion) {
        this.des_idordenatencion = des_idordenatencion;
    }
}
```

Fuente: Elaboración propia

En la figura 145 se puede ver el código del modelo descanso médico, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase DescansoMedicoModel.

Figura 146 Vista Descanso

```
$.ajax({
    type: (op.length == 0 ? "POST" : "PUT"), //SI TIENE ES UNA ACTUALIZACION
    url: "${pageContext.request.contextPath}/DescansoMedico/RegisUpd",
    crossDomain: true,
    data: formToJson(),
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    success: function (response) {
        if(response.status == true){
            $.toast({
                heading: 'Exito',
                text: 'DESCANSO MEDICO REGISTRADO SATISFACTORIAMENTE',
                position: 'bottom-right',
                icon: 'success'
            });
            LimpiarTxtDescansoMedico();
            $("#formMantDescansoMedico")[0].reset();
            $("#ContenDescansoMedico").load('../DescansoMedico/WebEmergencia');
        }else{
            $.toast({
                heading: 'Error',
                text: 'NO SE PUDO REGISTRAR ',
                position: 'bottom-right',
                icon: 'error'
            });
        }
    },
});
```

Fuente: Elaboración propia

En la figura 146 se puede ver un fragmento del código de la vista de gestionar descanso médico el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 147 Controlador Descanso Medico

```
@RequestMapping(value="/RegisUpd",method=RequestMethod.POST,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getRegistrarDescansoMedico(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }

        DescansoMedicoBean descansoMedicoBean = (DescansoMedicoBean) new JsonSerializer().deserialize(jsonEnviado.toString(),DescansoMedicoBean.class);
        result = descansoMedicoService.RegistrarDescMedico(descansoMedicoBean);

        mapaResult.put(Constantes.STATUS_TEXT, result);
        mapaResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constantes.STATUS_TEXT, false);
        mapaResult.put(Constantes.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}

@RequestMapping(value="/RegisUpd",method=RequestMethod.PUT,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getActualizarDescansoMedico(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }

        DescansoMedicoBean descansoMedicoBean = (DescansoMedicoBean) new JsonSerializer().deserialize(jsonEnviado.toString(),DescansoMedicoBean.class);
        result = descansoMedicoService.ActualizarDescansoMedico(descansoMedicoBean);

        mapaResult.put(Constantes.STATUS_TEXT, result);
        mapaResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constantes.STATUS_TEXT, false);
        mapaResult.put(Constantes.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
}
```

Fuente: Elaboración propia

En la figura 147 se puede ver el código del controlador descanso médico para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de registro de descanso médico o mensaje de error.

Figura 148 Ejecución BD Registra Descanso Medico

```
@Override
public boolean InsertarDescansoMedico(DescansoMedicoBean descansoMedicoBean) throws Exception {
    // TODO Auto-generated method stub
    if(Log.isDebugEnabled()) Log.debug("Inicio - DescansoMedicoDAOImpl.InsertarDescansoMedico");
    String SQL = "EXEC SP_TR_INSERT_DESCANSO_MEDICO ?,?,?,?,?";

    try {
        rows= jdbcTemplate.update(SQL.toString(),new Object[] {

            descansoMedicoBean.getDes_idordenatencion(),
            descansoMedicoBean.getDes_fechaactual(),
            descansoMedicoBean.getDes_observacion(),
            descansoMedicoBean.getDes_f_del(),
            descansoMedicoBean.getDes_f_al(),
            descansoMedicoBean.getDes_diasdescanso(),
            descansoMedicoBean.getDes_usuarioCreacion()

        });
        System.out.print(""+descansoMedicoBean.getDes_idordenatencion());
    } catch (Exception ex) {
        if(Log.isDebugEnabled()) Log.debug("Inicio - DescansoMedicoDAOImpl.InsertarDescansoMedico");
        Log.error(ex,ex);
        throw ex;
        // TODO: handle exception
    }finally {
        if(Log.isDebugEnabled()) Log.debug("Inicio - DescansoMedicoDAOImpl.InsertarDescansoMedico");
    }
    return rows==1;
}
```

Fuente: Elaboración propia

En la figura 148 se puede ver el código del modelo de datos para registrar descanso médico y su ejecución en la base de datos para luego devolver la respuesta.

Figura 149 Modelo Alta Medica

```
package pe.edu.sanna.triage.model;

public abstract class AltaMedicaModel{

    int idAltaMedica;
    String description;

    public int getIdAltaMedica() {
        return idAltaMedica;
    }
    public void setIdAltaMedica(int idAltaMedica) {
        this.idAltaMedica = idAltaMedica;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

Fuente: Elaboración propia

En la figura 149 se puede ver el código de la modelo alta médica, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase AltaMedicalModel.

Figura 150 Vista Alta Medica

```
$.ajax({
    type: "PUT",
    url: "${pageContext.request.contextPath}/OrdenAtencion/AltaMediEmergencia",
    crossDomain: true,
    data: AltaToJSON,
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    success: function (response) {

        if(response.status === true){

            $.toast({
                heading: 'Exito',
                text: 'ALTA MÉDICA SATISFACTORIA',
                position: 'top-center',
                icon: 'success'
            });

            LoadDataTablesAjax();

            document.getElementById('txtContrase').value = '';
            document.getElementById('txtMensaje').value = '';

            $('#exampleModalCenter').modal('hide');

        }else{
            //alert("Validar Credenciales o debe tener un Diagnostico y alta de Enfermeria");
            $.toast({
                heading: 'Error',
                text: 'VALIDAR CREDENCIALES O AUN FALTA REGISTRAR DIAGNOSTICO Y/O ACTO DE ENFERMERIA EN ATENDIDO',
                position: 'top-center',
                icon: 'error'
            });

            document.getElementById('txtContrase').value = '';
            document.getElementById('txtMensaje').value = '';

        }
    }
});
```

Fuente: Elaboración propia

En la figura 150 se puede ver un fragmento del código de la vista de alta médica el cual se está enviando por Ajax mediante una ruta al controlador los atributos a actualizar al estado atendido para poder realizar la ejecución con la BD.

Figura 151 Controlador Alta Medica

```
@RequestMapping(value="/AltaMediEmergencia",method=RequestMethod.PUT,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept-application/json")
public @ResponseBody Map<String, Object> getAltaMediEmergencia(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }

        OrdenAtencionBean orden = (OrdenAtencionBean) new JsonSerializer().deserialize(jsonEnviado.toString(), OrdenAtencionBean.class);

        // ACTUALIZAR USUARIO
        result = ordenAtencionService.darAltaMedica(orden);

        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_UPDATE.replace("{0}", (result ? "1": "0")));

    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 151 se puede ver el código del controlador alta médica para poder realizar el envío a la BD para la validación de datos y obtener la respuesta de actualización de estado atendido o mensaje de error.

Figura 152 Ejecución BD Alta Medica

```
@Override
public boolean darAltaMedica(OrdenAtencionBean ordenAtencionBean) throws Exception {
    // TODO Auto-generated method stub
    if (Log.isDebugEnabled()) Log.debug("Inicio - OrdenAtencionDAOImpl.darAltaMedica");

    String SQL = "EXEC SP_ALTA_MEDICA_EMERGENCIA ?,?,?,?,?";

    try {
        rows = jdbcTemplate.update(SQL, new Object[] {
            ordenAtencionBean.getIdordenAtencion(),
            ordenAtencionBean.getIdUsuarioalta(),
            ordenAtencionBean.getPass(),
            ordenAtencionBean.getIdAltaMedica(),
            ordenAtencionBean.getObservacion_alta(),
            ordenAtencionBean.getArchivo()
        });

    } catch (Exception ex) {
        // TODO: handle exception
        if (Log.isDebugEnabled()) Log.debug("Error - OrdenAtencionDAOImpl.darAltaMedica");
        Log.error(ex, ex);
        throw ex;
    } finally {
        if (Log.isDebugEnabled()) Log.debug("Final - OrdenAtencionDAOImpl.darAltaMedica");
    }
    return rows==1;
}
```

Fuente: Elaboración propia

En la figura 152 se puede ver el código del modelo de datos para actualizar estado atendido de atención y su ejecución en la base de datos para luego devolver la respuesta.

Figura 153 Modelo Nivel Percentil de Cumplimiento

```
package pe.edu.sanna.triage.modelo;

public abstract class ItemNivCumplimientoReporteModel {

    int item;
    String fecha;
    int usuario;
    String turno;
    int cpac;
    int p1;
    int p2;
    int p3;
    int p4;
    int p5;
    double npc;
    public int getItem() {
        return item;
    }
    public void setItem(int item) {
        item = item;
    }
    public String getFecha() {
        return fecha;
    }
    public void setFecha(String fecha) {
        this.fecha = fecha;
    }
    public int getUsuario() {
        return usuario;
    }
    public void setUsuario(int usuario) {
        this.usuario = usuario;
    }
    public String getTurno() {
        return turno;
    }
    public void setTurno(String turno) {
        this.turno = turno;
    }
    public int getCpac() {
        return cpac;
    }
    public void setCpac(int cpac) {
        this.cpac = cpac;
    }
    public int getP1() {
        return p1;
    }
    public void setP1(int p1) {
        this.p1 = p1;
    }
    public int getP2() {
        return p2;
    }
    public void setP2(int p2) {
        this.p2 = p2;
    }
}
```

Fuente: Elaboración propia

En la figura 153 se puede ver el código del modelo nivel percentil de cumplimiento, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase ItemNivCumplimientoModel.

Figura 154 Vista Nivel Percentil de Cumplimiento

```
function LoadDataTableWithAjax(){
    var periodo=document.getElementById('txtTemas').value;

    $('#data-table-default').dataTable().fnClearTable();
    $('#data-table-default').dataTable().fnDraw();
    $.ajax({
        type: "GET",
        url: "${pageContext.request.contextPath}/ItemNivelCumplReport/ListNivelCum/"+periodo,
        crossDomain: true,
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        async: true,
        success: function (response) {
            //ADD NOTIFICATION
            //ShowMessage(response.message);

            if(response.data.length > 0){
                //UPDATE DATATABLES
                $('#data-table-default').dataTable().fnAddData(response.data);
                //alert(JSON.stringify(response.data));
            }else{
                console.log("No hay Atenciones")
            }
            //UPDATE DATATABLES
        }
    });
    //Error: OnErrorCall
}

$("#btnBuscar").click(function() {
    // $('#modal-dialog').modal('show');
    // $('#stack1').modal('show');
    var periodo=document.getElementById('txtTemas').value;

    if(periodo==""){
        $.toast({
            heading: 'warning',
            text: 'No ha SELECCIONADO PERIODO',
            position: 'top-center',
            showHideTransition: 'plain',
            icon: 'warning'
        });
    }
    LoadDataTableWithAjax()
});
```

Fuente: Elaboración propia

En la figura 154 se puede ver un fragmento del código de la vista de nivel percentil de cumplimiento el cual se está enviando por Ajax mediante una ruta al controlador los atributos a listar para poder realizar la ejecución con la BD.

Figura 155 Controlador Nivel Percentil de Cumplimiento

```
@RequestMapping(value="/ListNivelCum/{periodo}",method=RequestMethod.GET,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> listarNivelCumReporte(@PathVariable String periodo) throws Exception {

    System.out.print("ENTRO");
    Map<String, Object>mapResult=new HashMap<>();

    try {

        List<ItemNivCumplimientoReporteBean> lstItemReport =itemNivCumReportService.listarNivelCumReporte(periodo);

        mapResult.put(Constantes.STATUS_TEXT, true);
        mapResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_READ.replace("{0}", String.valueOf(lstItemReport.size())));
        mapResult.put("data", lstItemReport);

        System.out.print("encontrado: "+lstItemReport.size());

    } catch (Exception e) {
        Log.error("ItemNivCumReportController.listarNivelCumReporte - Error");
        Log.error(e.getMessage());
        mapResult.put(Constantes.STATUS_TEXT, false);
        mapResult.put(Constantes.MESSAGE_TEXT, e.getLocalizedMessage());
        // TODO: handle exception
    }
    return mapResult;
}
```

Fuente: Elaboración propia

En la figura 155 se puede ver el código del controlador nivel percentil de cumplimiento para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de listar nivel percentil de cumplimiento o mensaje de error.

Figura 156 Ejecución BD Nivel Percentil de Cumplimiento

```
@Override
public List<ItemNivCumplimientoReporteBean> listarNivelCumReport(String periodo) throws Exception {
    // TODO Auto-generated method stub
    if (Log.isDebugEnabled()) Log.debug("Inicio - ItemNivCumReportDAOImpl.listarNivelCumReport");
    List<ItemNivCumplimientoReporteBean> lstOrden;
    String SQL = "EXEC SP_LISTAR_NIVEL_PERCENTIL_CUMPLIMIENTO ?";
    try {

        lstOrden = jdbcTemplate.query(SQL, new Object[] { periodo }, new ResultSetExtractor<List<ItemNivCumplimientoReporteBean>>() {
            @Override
            public List<ItemNivCumplimientoReporteBean> extractData(ResultSet rs) throws SQLException, DataAccessException {
                // TODO Auto-generated method stub
                List<ItemNivCumplimientoReporteBean> lista = new ArrayList<>();
                while (rs.next()) {
                    ItemNivCumplimientoReporteBean orden= new ItemNivCumplimientoReporteBean();

                    orden.setItem(rs.getInt("item"));
                    orden.setFecha(rs.getString("fecha"));
                    orden.setUsuario(rs.getInt("usuario"));
                    orden.setTurno(rs.getString("turno"));
                    orden.setCpac(rs.getInt("cpac"));
                    orden.setP1(rs.getInt("p1"));
                    orden.setP2(rs.getInt("p2"));
                    orden.setP3(rs.getInt("p3"));
                    orden.setP4(rs.getInt("p4"));
                    orden.setP5(rs.getInt("p5"));
                    orden.setNpc(rs.getDouble("npc"));

                    lista.add(orden);
                }
                return lista;
            }
        });

    } catch (Exception ex) {
        // TODO: handle exception
        if (Log.isDebugEnabled()) Log.debug("Error - ItemNivCumReportDAOImpl.listarNivelCumReport");
        Log.error(ex, ex);
        throw ex;
    } finally {
        if (Log.isDebugEnabled()) Log.debug("Final - ItemEmergenciaDAOImpl.listarReporteGeneral");
    }
    return lstOrden;
}
```

Fuente: Elaboración propia

En la figura 156 se puede ver el código del modelo de datos para listar nivel percentil de cumplimiento y su ejecución en la base de datos para luego devolver la respuesta.

**Figura 157 Modelo Reporte
Emergencia**

```
package pe.edu.sanna.triage.model;

public abstract class ItemEmergencia {

    int historiaclinica;
    String seguro;
    String nombrecompleto;
    String estadoemergencia;
    String medico;
    String prioridad;
    String topicoderivar;
    String motivo;
    String diagnostico1;
    String diagnostico2;
    String diagnostico3;
    String fecharegistrotriage;
    String fechaaperturaemergencia;
    String fechaaltaemergencia;
    String alta;
    public int getHistoriaclinica() {
        return historiaclinica;
    }
    public void setHistoriaclinica(int historiaclinica) {
        this.historiaclinica = historiaclinica;
    }
    public String getSeguro() {
        return seguro;
    }
    public void setSeguro(String seguro) {
        this.seguro = seguro;
    }
    public String getNombrecompleto() {
        return nombrecompleto;
    }
    public void setNombrecompleto(String nombrecompleto) {
        this.nombrecompleto = nombrecompleto;
    }
    public String getEstadoemergencia() {
        return estadoemergencia;
    }
    public void setEstadoemergencia(String estadoemergencia) {
        this.estadoemergencia = estadoemergencia;
    }
    public String getMedico() {
        return medico;
    }
    public void setMedico(String medico) {
        this.medico = medico;
    }
    public String getPrioridad() {
        return prioridad;
    }
    public void setPrioridad(String prioridad) {
        this.prioridad = prioridad;
    }
}
```

Fuente: Elaboración propia

En la figura 157 se puede ver el código del modelo reporte emergencia, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase ItemReportEmergenciaModel.

**Figura 158 Vista Reporte
Emergencia**

```
function LoadDataTablesWithAjax(){
    var periodo=document.getElementById('txtTopicos').value;

    $('#data-table-default').dataTable().fnClearTable();
    $('#data-table-default').dataTable().fnDraw();
    $.ajax({
        type: "GET",
        url: "${pageContext.request.contextPath}/ItemEmergencia/ListReportEmergencia/"+periodo,
        crossDomain: true,
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        async: true,
        success: function (response) {
            //ADD NOTIFICACION
            //ShowMessage(response.message);

            if(response.data.length > 0){
                //UPDATE DATATABLES
                $('#data-table-default').dataTable().fnAddData(response.data);
                //alert(JSON.stringify(response.data));

            }else{
                console.log("No hay Atenciones")
            }

            //UPDATE DATATABLES

        },
        //,
        //error: OnErrorCall
    })
}

$( "#btnBuscar" ).click(function() {
    // $('#modal-dialog').modal('show');
    //$('#stack1').modal('show')
    var periodo=document.getElementById('txtTopicos').value;

    if(periodo=="0"){
        $.toast({
            heading: 'Warning',
            text: 'NO HA SELECCIONADO PERIODO',
            position: 'top-center',
            showHideTransition: 'plain',
            icon: 'warning'
        });
    }
    LoadDataTablesWithAjax()
})
```

Fuente: Elaboración propia

En la figura 158 se puede ver un fragmento del código de la vista de reporte emergencia el cual se está enviando por Ajax mediante una ruta al controlador los atributos a listar para poder realizar la ejecución con la BD.

Figura 159 Controlador Reporte Emergencia

```
@RequestMapping(value="/ListReporteEmergencia/{periodo}",method=RequestMethod.GET,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> listarReporteGeneral(@PathVariable String periodo) throws Exception {

    Map<String, Object>mapResult=new HashMap<>();

    try {
        List<ItemEmergenciaBean> lstItemRepot =itemEmergenciaService.listarReporteGeneral(periodo);
        mapResult.put(Constants.STATUS_TEXT, true);
        mapResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_READ.replace("{0}", String.valueOf(lstItemRepot.size())));
        mapResult.put("data", lstItemRepot);
    } catch (Exception e) {
        Log.error("ItemEmergenciaController.listarReporteGeneral - Error");
        Log.error(e.getMessage());
        mapResult.put(Constants.STATUS_TEXT, false);
        mapResult.put(Constants.MESSAGE_TEXT, e.getLocalizedMessage());
        // TODO: handle exception
    }
    return mapResult;
}
```

Fuente: Elaboración propia

En la figura 159 se puede ver el código del controlador reporte emergencia para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de listar reporte emergencia o mensaje de error.

Figura 160 Ejecución BD Reporte Emergencia

```
@Override
public List<ItemEmergenciaBean> listarReporteGeneral(String periodo) throws Exception {
    if (Log.isDebugEnabled()) Log.debug("Inicio - ItemEmergenciaDAOImpl.listarReporteGeneral");
    List<ItemEmergenciaBean> lstOrden;
    String SQL ="EXEC SP_LISTAR_REPORTES EMERGENCIA ?";
    try {

        lstOrden = jdbcTemplate.query(SQL, new Object[] {periodo}, new ResultSetExtractor<List<ItemEmergenciaBean>>() {
            @Override
            public List<ItemEmergenciaBean> extractData(ResultSet rs) throws SQLException, DataAccessException {
                List<ItemEmergenciaBean> lista = new ArrayList<>();
                while (rs.next()) {
                    ItemEmergenciaBean orden=new ItemEmergenciaBean();
                    orden.setHistoriaclinica(rs.getInt("historiaclinica"));
                    orden.setSeguro(rs.getString("seguro"));
                    orden.setNombrecompleto(rs.getString("nombrecompleto"));
                    orden.setEstadoemergencia(rs.getString("estadoemergencia"));
                    orden.setMedico(rs.getString("medico"));
                    orden.setPrioridad(rs.getString("prioridad"));
                    orden.setTopicoderivar(rs.getString("topicoderivar"));
                    orden.setDiagnostico1(rs.getString("diagnostico1"));
                    orden.setDiagnostico2(rs.getString("diagnostico2"));
                    orden.setDiagnostico3(rs.getString("diagnostico3"));
                    orden.setFecharegistrotriage(rs.getString("fecharegistrotriage"));
                    orden.setFechaaperturaemergencia(rs.getString("fechaaperturaemergencia"));
                    orden.setFechaaltaemergencia(rs.getString("fechaaltaemergencia"));
                    orden.setAlta(rs.getString("alta"));

                    lista.add(orden);
                }
                return lista;
            }
        });
    } catch (Exception ex) {
        // TODO: handle exception
        if (Log.isDebugEnabled()) Log.debug("Error - ItemEmergenciaDAOImpl.listarReporteGeneral");
        Log.error(ex, ex);
        throw ex;
    } finally {
        if (Log.isDebugEnabled()) Log.debug("Final - ItemNivCumpReportDAOImpl.listarNivelCumpReport");
    }
    return lstOrden;
}
```

Fuente: Elaboración propia

En la figura 160 se puede ver el código del modelo de datos para listar el reporte de emergencia y su ejecución en la base de datos para luego devolver la respuesta.

Figura 161 Modelo Nota Enfermería

```
package pe.edu.sanna.triage.model;

public abstract class NotaEnfermeriaModel {

    int enf_idordenatencion;
    String fecha_nota;
    String nota_enfermeria;
    String enf_usuarioCreacion;

    public int getEnf_idordenatencion() {
        return enf_idordenatencion;
    }
    public void setEnf_idordenatencion(int enf_idordenatencion) {
        this.enf_idordenatencion = enf_idordenatencion;
    }
    public String getFecha_nota() {
        return fecha_nota;
    }
    public void setFecha_nota(String fecha_nota) {
        this.fecha_nota = fecha_nota;
    }
    public String getNota_enfermeria() {
        return nota_enfermeria;
    }
    public void setNota_enfermeria(String nota_enfermeria) {
        this.nota_enfermeria = nota_enfermeria;
    }
    public String getEnf_usuarioCreacion() {
        return enf_usuarioCreacion;
    }
    public void setEnf_usuarioCreacion(String enf_usuarioCreacion) {
        this.enf_usuarioCreacion = enf_usuarioCreacion;
    }
}
```

Fuente: Elaboración propia

En la figura 161 se puede ver el código del modelo nota enfermería, en donde se puede ver la declaración de los get's y set's pertenecientes a la clase NotaEnfermeriaModel.

Figura 162 Vista Nota Enfermería

```
$.ajax({
    type: (op.length == 0 ? "POST" : "PUT"), //SI TIENE ES UNA ACTUALIZACION
    url: "${pageContext.request.contextPath}/NotasEnfermeria/RegisUpd",
    crossDomain: true,
    data: formToJson(),
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    success: function (response) {

        if(response.status == true){

            $.toast({
                heading: 'Exito',
                text: 'NOTA ENFERMERIA REGISTRADO SATISFACTORIAMENTE',
                position: 'bottom-right',
                icon: 'success'
            });

            LimpiarTxtNota();
            $("#formNantNotaEnfermeria")[0].reset();
            //location.reload();
            $("#Contnotas").load('../NotasEnfermeria/WebEnfermeria');
        }else{

            $.toast({
                heading: 'Error',
                text: 'NO SE PUEDO REGISTRAR',
                position: 'bottom-right',
                icon: 'error'
            });

        }

    },
});
```

Fuente: Elaboración propia

En la figura 162 se puede ver un fragmento del código de la vista de gestionar nota enfermería el cual se está enviando por Ajax mediante una ruta al controlador los atributos a registrar para poder realizar la ejecución con la BD.

Figura 163 Controlador Nota Enfermería

```
@RequestMapping(value="/RegisUpd",method=RequestMethod.POST,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getRegistrarNotasEnfermeria(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }
        NotaEnfermeriaBean notaEnfermeriaBean = (NotaEnfermeriaBean) new JsonSerializer().deserialize(jsonEnviado.toString(),NotaEnfermeriaBean.class);

        result = notasEnfermeriaService.registrarNotasEnfermeria(notaEnfermeriaBean);

        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_INSERT.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}

// METODO PUT ACTUALIZAR UN REGISTRO <- JSON
@RequestMapping(value="/RegisUpd", method = RequestMethod.PUT, produces = MediaType.APPLICATION_JSON_VALUE, headers = "Accept=application/json")
public @ResponseBody Map<String, Object> ActualizarNotaEnfermeria(HttpServletRequest request, HttpServletResponse response) throws Exception {
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }

        NotaEnfermeriaBean Upnota = (NotaEnfermeriaBean) new JsonSerializer().deserialize(jsonEnviado.toString(), NotaEnfermeriaBean.class);

        // ACTUALIZAR USUARIO
        result = notasEnfermeriaService.actualizarNotasEnfermeria(Upnota);
        mapaResult.put(Constants.STATUS_TEXT, result);
        mapaResult.put(Constants.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_UPDATE.replace("{0}", (result ? "1": "0")));
    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constants.STATUS_TEXT, false);
        mapaResult.put(Constants.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 163 se puede ver el código del controlador nota enfermería para poder realizar él envío a la BD para la validación de datos y obtener la respuesta de registro de nota enfermería o mensaje de error.

Figura 164 Ejecución BD Nota Enfermería

```
@Override
public boolean registrarNotasEnfermeria(NotaEnfermeriaBean notasEnfermeriaBean) throws Exception {
    // TODO Auto-generated method stub
    if(Log.isDebugEnabled()) Log.debug("Inicio - NotaEnfermeriaDAOImpl.registrarNotasEnfermeria");
    String SQL = "EXEC SP_TR_INSERT_NOTAS_ENFERMERIA ?,?,?,?" ;
    try {
        //YA AHORA SI EJECUTALO

        rows= jdbcTemplate.update(SQL.toString(),new Object[] {
            notasEnfermeriaBean.getEnf_idordenatencion(),
            notasEnfermeriaBean.getFecha_nota(),
            notasEnfermeriaBean.getNota_enfermeria(),
            notasEnfermeriaBean.getEnf_usuarioCreacion()
        });

    } catch (Exception ex) {
        if(Log.isDebugEnabled()) Log.debug("Inicio - NotaEnfermeriaDAOImpl.registrarNotasEnfermeria");
        Log.error(ex,ex);
        throw ex;
    } finally {
        // TODO: handle exception
        if(Log.isDebugEnabled()) Log.debug("Inicio - NotaEnfermeriaDAOImpl.registrarNotasEnfermeria");
    }
    return rows==1;
}

//LISTAR PERSONA POR DOCUMENTO
```

Fuente: Elaboración propia

En la figura 164 se puede ver el código del modelo de datos para registrar nota enfermería y su ejecución en la base de datos para luego devolver la respuesta.

Figura 165 Vista Acto Enfermería

```
$( "#btnatendido" ).click(function() {

    $.ajax({
        type: "PUT",
        url: "${pageContext.request.contextPath}/OrdenAtencion/AtendidoEnfermeria",
        crossDomain: true,
        data: formToJSON,
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        async: true,
        success: function (response) {
            if(response.status === true){
                var d = response.data;

                //alert("SATISFACTO");

                $.toast({
                    heading: 'Exito',
                    text: 'ACTO ENFERMERIA SATISFACTORIA',
                });

                LoadDataTableWithAjax();

            }else{
                // alert("NO POSEE NOTAS DE ENFERMERIA");
                $.toast({
                    heading: 'Error',
                    text: 'NO HA REGISTRADO NOTAS ENFERMERIA',
                    position: 'top-center',
                    icon: 'error'
                });
            }
        },
    });
});
```

Fuente: Elaboración propia

En la figura 165 se puede ver un fragmento del código de la vista de bandeja enfermería el cual se está enviando por Ajax mediante una ruta al controlador los atributos a actualizar al estado atendido para poder realizar la ejecución con la BD.

Figura 166 Controlador Acto Enfermería

```
@RequestMapping(value="/AtendidoEnfermeria",method=RequestMethod.PUT,produces=MediaType.APPLICATION_JSON_VALUE,headers = "Accept=application/json")
public @ResponseBody Map<String, Object> getActualizarEstadoAtendidoEnfermeria(HttpServletRequest request,HttpServletResponse response) throws Exception{
    Map<String, Object> mapaResult = new HashMap<>();
    try {
        //LECTURA DEL REQUEST BODY
        BufferedReader reader = new BufferedReader(new InputStreamReader(request.getInputStream(),"UTF-8"));
        StringBuilder jsonEnviado = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonEnviado.append(line).append('\n');
        }

        OrdenAtencionBean orden = (OrdenAtencionBean) new JsonSerializer().deserialize(jsonEnviado.toString(), OrdenAtencionBean.class);

        // ACTUALIZAR USUARIO
        result = ordenAtencionService.actualizarEstadoAtendidoEnfermeria(orden);

        mapaResult.put(Constantes.STATUS_TEXT, result);
        mapaResult.put(Constantes.MESSAGE_TEXT, AvisosConstantes.AVISO_CRUD_UPDATE.replace("{0}", (result ? "1": "0")));

    } catch (Exception ex) {
        ex.printStackTrace();
        mapaResult.put(Constantes.STATUS_TEXT, false);
        mapaResult.put(Constantes.MESSAGE_TEXT, ex.getLocalizedMessage());
    }
    return mapaResult;
}
```

Fuente: Elaboración propia

En la figura 166 se puede ver el código del controlador acto médico para poder realizar el envío a la BD para la validación de datos y obtener la respuesta de actualización de estado atendido o mensaje de error.

Figura 167 Ejecución BD Acto Enfermería

```
@Override
public boolean actualizarEstadoAtendidoEnfermeria(OrdenAtencionBean ordenAtencionBean) throws Exception {
    // TODO Auto-generated method stub
    if (Log.isDebugEnabled()) Log.debug("Inicio - OrdenAtencionDAOImpl.actualizarEstadoAtendidoEnfermeria");

    String SQL = "exec SP_ACTUALIZAR_ESTADO_ATENDIDO_ENFERMERIA ?,?";

    try {
        rows = jdbcTemplate.update(SQL, new Object[] {
            ordenAtencionBean.getIdordenAtencion(),
            ordenAtencionBean.getUsuario_regenfermeria()
        });

        System.out.println("idUsuario:" + ordenAtencionBean.getUsuario_regenfermeria());
    } catch (DataAccessException e) {

        e.printStackTrace();
        SQLException sqle = (SQLException) e.getCause();
        System.out.println("MENSAJE DEVOLUCION SQL" + sqle.getMessage());

        /*
        System.out.println("Error code: " + sqle.getErrorCode());
        System.out.println("SQL state: " + sqle.getSQLState());*/

        throw e;
    } catch (Exception e) {
        e.printStackTrace();
        throw e;
    }

    return rows==1;
}
```

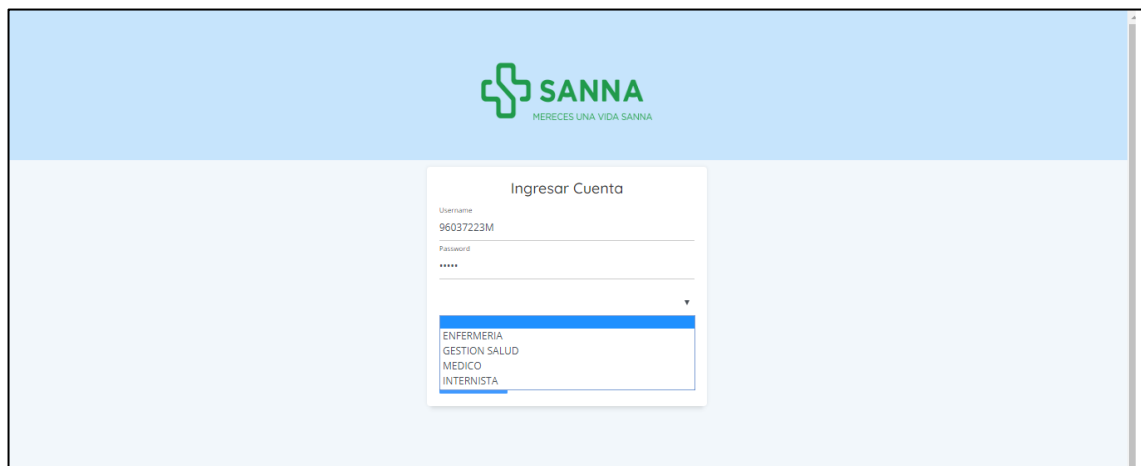
Fuente: Elaboración propia

En la figura 167 se puede ver el código del modelo de datos para actualizar estado atendido de atención y su ejecución en la base de datos para luego devolver la respuesta.

Implementación

Luego de la codificación de cada caso de uso, se procedió a ejecutar el sistema en un navegador. Los resultados obtenidos fueron los siguientes.

Figura 168 Interfaz de Ingresar al Sistema



Fuente: Elaboración propia

En la Figura 168, el sistema web nos muestra la pantalla de ingresar al sistema en donde nos solicitara el usuario, contraseña y tipo de usuario, luego dar clic en “Ingresar al sistema”.

Figura 169 Interfaz Gestionar Usuario

Figura 169 Interfaz Gestionar Usuario

Fuente: Elaboración propia

En la Figura 169 el sistema web nos carga la pantalla de gestión de usuario, el cual nos permite el registro y actualización de usuarios.

Figura 170 Interfaz Gestionar Paciente

Figura 170 Interfaz Gestionar Paciente

Fuente: Elaboración propia

En la Figura 170 el sistema web nos carga la pantalla de gestión de paciente, el cual nos permite el registro y actualización de pacientes.

Figura 171 Interfaz Crear Orden Atención

Fuente: Elaboración propia

En la Figura 171 el sistema web nos carga la pantalla de crear orden atención en donde iniciamos buscando el dni de paciente, luego seleccionamos y detallamos el ingreso del paciente, finalizando dando cli en “crear orden”.

Figura 172 Interfaz Listar Orden Atención en Bandeja Admisionista

Fuente: Elaboración propia

En la Figura 172 el sistema nos carga la bandeja admisionista en donde listara las ordenes de atención que se encuentran en atención de triaje en los estados de pendiente y en atención.

Figura 173 Interfaz Listar Orden Atención en Bandeja Internista

Fuente: Elaboración propia

En la Figura 173 el sistema nos carga la bandeja internista en donde listara las ordenes de atención que se encuentran en espera de triaje en los estados de pendiente y en atención. Además, nos permite seleccionar una fila de la bandeja internista para dar clic en “abrir atención” para aperturar la atención de triaje o clic en “abandono” para actualizar el estado de la orden a abandono.

Figura 174 Interfaz Listar Orden Atención en Bandeja Medico

Fuente: Elaboración propia

En la Figura 174 el sistema nos carga la bandeja de médico en donde listara las ordenes de atención que se encuentran en espera de atención de emergencia en los estados de pendiente, en atención y atendido. Además, nos permite seleccionar una fila de la bandeja médico para dar clic en “abrir atención” para aperturar la atención de emergencia o clic en “abandono” para actualizar el estado de la orden a abandono.

Figura 175 Interfaz Listar Orden Atención en Bandeja Enfermera

Fuente: Elaboración propia

En la Figura 175 el sistema nos carga la bandeja enfermera en donde listara las ordenes de atención que se encuentran en espera de atención de enfermería en los estados de pendiente, en atención y atendido. Además, nos permite seleccionar una fila de la bandeja enfermera para dar clic en “abrir atención” para aperturar la atención de enfermería o clic en “abandono” para actualizar el estado de la orden a abandono.

Figura 176 Interfaz Motivos de Atención

Fuente: Elaboración propia

En la Figura 176 el sistema nos carga la pantalla de motivos de atención apenas aperturamos una atención en triaje, en esta pantalla seleccionamos un motivo de ingreso o dolencia característica del relato del paciente.

Figura 177 Interfaz Registrar Triage

Motivos de Atención

Detalles Paciente y Signos Vitales

Selecciona Tema...

Prioridad:

Fecha Ingreso: 28/11/2019

Hora Ingreso: 23:34:30

Edad Paciente: 47

Sexo Paciente: Femenino

Paciente: URBUNAGA RUIZ, DOBRO DOBRO

Motivo de Ingreso:

Atmá:

Discriminador Prioridad 1

- ☐ vía aérea comprometida
- ☐ respirar inadecuadamente
- ☐ shock
- ☐ niño que no responde

Discriminador Prioridad 2

- ☐ incapaz de decir frases
- ☐ saturación de oxígeno muy baja
- ☐ flujo respiratorio en pico muy bajo
- ☐ pulso anormalmente
- ☐ nivel de conciencia alterado
- ☐ historia respiratoria significativa
- ☐ hipertermia

Discriminador Prioridad 3

- ☐ saturación de O2 baja
- ☐ flujo respiratorio en pico bajo
- ☐ asma que no mejora con el propio tratamiento
- ☐ fiebre alta

Discriminador Prioridad 4

- ☐ sibilancia
- ☐ tos productiva
- ☐ problema reciente

Discriminador Prioridad 5

- ☐ sin discriminador

☐ Habiendo FUR ☐ Alergia

SIGNOS VITALES

Peso: KG PA: mmHg T°: SAT O2: % F.C.: /min F.R.: /min

Antecedentes

Observación

Fuente: Elaboración propia

En la Figura 177 el sistema nos carga la pantalla de registro de triaje en donde podemos seleccionar uno o más discriminadores para brindar la prioridad de atención en emergencia, además podremos registrar signos vitales y seleccionar un tópico de emergencia.

Figura 178 Interfaz Reporte Nivel Porcentual de Pacientes Atendidos en Triage

INTERISTA: JIMENEZ GONZALES, MARIA NATALIT 09079214-J

NIVEL PORCENTUAL DE PACIENTES ATENDIDOS EN TRIAGE

Selecciona Periodo...

Buscar:

Item	Ti	Fecha	Ti	Internista	Ti	Turno	Ti	CTP	Ti	CPC	Ti	CPA
No hay registros disponibles												

Showing 0 to 0 of 0 entries

Fuente: Elaboración propia

En la Figura 178 el sistema web nos carga la pantalla de reporte nivel porcentual de pacientes atendidos en triage en donde seleccionaremos el mes a consultar para cargar la data y luego damos clic en “Exportar Excel” para mostrarnos la data en formato Excel.

Figura 179 Interfaz Reporte Triage

Fuente: Elaboración propia

En la Figura 179 el sistema web nos carga la pantalla de reporte triage en donde seleccionaremos el mes a consultar para cargar la data y luego damos clic en “Exportar Excel” para mostrarnos la data en formato Excel.

Figura 180 Interfaz Enfermedad Actual

Fuente: Elaboración propia

En la Figura 180 el sistema nos carga la pantalla de enfermedad actual, en donde podemos registrar las características de la dolencia o enfermedad del paciente, también podemos modificar la información, además la data carga automáticamente al momento de dar clic en “registrar”.

Figura 181 Interfaz Funciones Vitales

HOJA TRIAGE
ENFERMEDAD ACTUAL
FUNCIONES VITALES
DIAGNOSTICO
EXAMENES DE APOYO
RECETA
DESCANSO MEDICO
NOTAS ENFERMERIA

FUNCIONES VITALES

PA: Ingresar PA / Ingresar PA2 mmHg
FC: Ingresar FC X min
RR: Ingresar RR X min
Temperatura: Ingresar Temperatura C
SATO2: Ingresar PA %
PesoKg: Ingresar Peso %
TallaCm: Ingresar Talla
IMC: Ingresar IMC

Registrar

Fuente: Elaboración propia

En la Figura 181 el sistema nos carga la pantalla de funciones vitales, en donde podemos registrar los datos vitales del paciente, también podemos modificar la información, además la data carga automáticamente al momento de dar clic en “registrar”.

Figura 182 Interfaz Diagnostico

DIAGNOSTICO

Síntomas y Signos Principales

Agregar Diagnósticos

Buscar:

IdDiagnostico	IdOrden	UsuCreador	Diagnostico	Determinacion Diagnostica	Grado Afeccion	Principal	Acción
No hay registros disponibles							

Registrar

Fuente: Elaboración propia

En la Figura 182 el sistema nos carga la pantalla de diagnóstico, en donde podemos agregar diagnósticos dando clic en “Agregar Diagnósticos” nos cargara una pantalla donde nos lista los distintos diagnósticos que se encuentran en la base de datos damos clic en el icono de “agregar” y nos cargara la información en la tabla de la pantalla, luego podemos registrar los datos agregados dando clic en “registrar”, también podemos modificar la información, además la data carga automáticamente al momento de dar clic en “registrar”, también podemos eliminar la fila del diagnóstico y al realizar esto la data carga automáticamente.

Figura 183 Interfaz Exámenes de Apoyo al Diagnostico

Fuente: Elaboración propia

En la Figura 183 el sistema nos carga la pantalla de exámenes de apoyo al diagnóstico, en donde podemos agregar exámenes dando clic en “Agregar Examen” nos cargara una pantalla donde nos lista los distintos exámenes que se encuentran en la base de datos damos clic en el icono de “agregar” y nos cargara la información en la tabla de la pantalla, luego podemos registrar los datos agregados dando clic en “registrar”, también podemos modificar la información, además la data carga automáticamente al momento de dar clic en “registrar”, también podemos eliminar la fila del examen y al realizar esto la data carga automáticamente.

Figura 184 Interfaz Receta

Fuente: Elaboración propia

En la Figura 184 el sistema nos carga la pantalla de receta, en donde podemos agregar medicamentos dando clic en “Agregar Medicamento” nos cargara una pantalla donde nos lista los distintos medicamentos que se encuentran en la base de datos damos clic en el icono de “agregar” y nos cargara la información en la tabla de la pantalla, luego podemos registrar los datos agregados dando clic en “registrar”, también podemos modificar la información, además la data carga automáticamente al momento de dar clic en “registrar”,

también podemos eliminar la fila del medicamento y al realizar esto la data carga automáticamente.

Figura 185 Interfaz descanso medico

Fuente: Elaboración propia

En la Figura 185 el sistema nos carga la pantalla de descanso médico, en donde podemos visualizar que carga el listado de diagnósticos agregados en la interfaz diagnósticos, también podemos registrar el tiempo de descanso del paciente o podemos modificar la información, además la data carga automáticamente al momento de dar clic en “registrar”.

Figura 186 Interfaz Reporte Nivel Percentil de Cumplimiento

Fuente: Elaboración propia

En la Figura 186 el sistema web nos carga la pantalla de reporte de nivel percentil de cumplimiento en donde seleccionaremos el mes a consultar para cargar la data y luego damos clic en “Exportar Excel” para mostrarnos la data en formato Excel.

Figura 187 Interfaz Reporte Emergencia

Fuente: Elaboración propia

En la Figura 187 el sistema web nos carga la pantalla de reporte emergencia en donde seleccionaremos el mes a consultar para cargar la data y luego damos clic en “Exportar Excel” para mostrarnos la data en formato Excel.

Figura 188 Interfaz alta medica

Fuente: Elaboración propia

En la Figura 188 en la pantalla bandeja medico al seleccionar una fila de la bandeja y al dar clic en “Alta Médica” nos carga la pantalla de alta médica en donde registramos la contraseña de usuario, seleccionamos un tipo de alta médica y si se requiere una observación, para finalizar damos clic en “Firmar”.

Figura 189 Reporte de atención emergencia

Acto medico 463 Nombre Paciente SHINNO HERNANDEZ , DANIEL DNI 01020702 ALFREDO Sexo Masculino Fecha-hora triaje 2019-11-13 12:25 Fecha-hora emergencia 2019-11-13 10:31		Resultado de atención ALTA MEDICA Historia Clínica 224 Edad 18 Seguro MAPFRE Prioridad atención PRIORIDAD II - EMERGENCIA Box de atención TOPICO 04													
Motivo de Ingreso Dolor toracico Signos vitales: Presión arterial (mmHg) Saturación de oxígeno Temperatura corporal (°C) Antropometría:		Tiempo de enfermedad 3Semanas Frecuencia cardiaca (latidos x Min) Frecuencia respiratoria x Min Peso (Kg) Talla (M) IMC													
Diagnóstico <table border="1"> <thead> <tr> <th>Diagnóstico</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>S20.2: CONTUSION DEL TORAX</td> <td>Principal</td> </tr> </tbody> </table>		Diagnóstico	Tipo	S20.2: CONTUSION DEL TORAX	Principal	Exámenes <table border="1"> <thead> <tr> <th>Examen</th> <th>Servicio</th> <th>Especificación</th> </tr> </thead> <tbody> </tbody> </table>		Examen	Servicio	Especificación					
Diagnóstico	Tipo														
S20.2: CONTUSION DEL TORAX	Principal														
Examen	Servicio	Especificación													
Receta <table border="1"> <thead> <tr> <th>Medicamento</th> <th>Dosis</th> <th>Cantidad</th> <th>Especificaciones</th> </tr> </thead> <tbody> <tr> <td>4061: TRAMADOL50mg/lny.</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>3086: KETOPROFENO2.5%Ge</td> <td>2</td> <td>2</td> <td>1</td> </tr> </tbody> </table>				Medicamento	Dosis	Cantidad	Especificaciones	4061: TRAMADOL50mg/lny.	1	1	1	3086: KETOPROFENO2.5%Ge	2	2	1
Medicamento	Dosis	Cantidad	Especificaciones												
4061: TRAMADOL50mg/lny.	1	1	1												
3086: KETOPROFENO2.5%Ge	2	2	1												
Médico: LARISCH FISCHMAN , JIMMY 31931															

Fuente: Elaboración propia

En la Figura 189 en la pantalla bandeja medico al seleccionar una fila de la bandeja y al dar clic en “Reporte” nos descargara un documento en formato pdf obteniendo un resumen de la atención de emergencia.

Figura 190 Interfaz Notas Enfermería

BANDEJA ENFERMERIA NOTAS ENFERMERIA	
Buscar: <input type="text"/>	
Diagnostico No hay registros disponibles	
Primero Anterior Siguiente Último	
Fecha Nota Enfermería 29/11/2019	
Nota Enfermería: <input type="text"/>	
Registrar	

Fuente: Elaboración propia

En la Figura 190 el sistema nos carga la pantalla de notas enfermería, en donde podemos visualizar que carga el listado de diagnósticos agregados en la interfaz diagnósticos en la atención médica, también podemos registrar el resumen de atención en enfermería y además podemos modificar la información, la data carga automáticamente al momento de dar clic en “registrar”.

Pruebas

Las pruebas realizadas en el sistema estuvieron enfocadas en probar su correcto funcionamiento, considerando criterios de validación, condición o algún evento dentro del formato que son esenciales para lograr el cumplimiento de los requerimientos previamente establecidos.

Tabla 89 Prueba Funcional Ingresar al Sistema

Condición /Evento / Campo	Clase válida	Clase no válida
Usuario	1.No puede quedar en blanco	2.En blanco
Contraseña	3.No puede quedar en blanco	4.En blanco
Tipo Usuario	5.Debe seleccionar un elemento	6.No debe seleccionar un elemento

Fuente: Elaboración propia

En la tabla 89 se ve los campos del caso de uso ingresar al sistema, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 90 Caso de Prueba Ingresar al Sistema

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2,4	No ingresa ningún dato	14/10/2109	Muestra mensajes “No ha ingresado usuario” o “No ha ingresado contraseña”	Satisfactoria

2	6	No selecciona elemento	14/10/2109	Muestra mensaje “No selecciono tipo de usuario”	Satisfactoria
3	1 o 3 o 5	Ingresa dato o selección de tipo usuario errónea	14/10/2109	Muestra mensaje “usuario, contraseña y/o tipo usuario erróneo”	Satisfactoria
4	1 o 3 o 5	Ingresa o selecciona un dato menos uno	14/10/2109	Muestra mensaje del campo a completar	Satisfactoria
5	1,3,5	Completa campos correctamente	14/10/2109	Ingresa al sistema	Satisfactoria

Fuente: Elaboración propia

En la tabla 90 se puede ver todas las pruebas realizadas al caso de ingresar al sistema, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 91 Prueba Funcional Gestionar Usuario

Condición /Evento / Campo	Clase válida	Clase no válida
Tipo usuario	1. Debe seleccionar un elemento. 3. Habilita campos según tipo usuario	2. No debe seleccionar elemento. 4. No habilita campos según tipo usuario.

Dni	<p>5. No puede quedar en blanco.</p> <p>7. Solo números.</p> <p>9. Permite solo 8 dígitos</p>	<p>6. En blanco.</p> <p>8. Ingresar letras</p> <p>10. Permite 8 dígitos o mas</p>
Nombre Completo	<p>11. No puede quedar en blanco.</p>	<p>12. En blanco.</p>
Fecha Nacimiento	<p>13. No puede quedar en blanco.</p> <p>15. No puede seleccionar fechas mayores a la actual.</p>	<p>14. En blanco.</p> <p>16. Selecciona fechas mayores a la actual.</p>
Sexo	<p>17. Debe seleccionar un elemento.</p>	<p>18. No debe seleccionar elemento.</p>
Correo Electrónico	<p>19. No puede quedar en blanco.</p> <p>21. Valida formato de correo.</p>	<p>20. En blanco.</p> <p>22. Permite ingresar cualquier tipo de valor.</p>
Estado Usuario	<p>23. Debe seleccionar un elemento.</p>	<p>24. No debe seleccionar elemento.</p>
Especialidad	<p>25. Debe seleccionar un elemento.</p>	<p>26. No debe seleccionar elemento.</p> <p>28. Habilitado para todos los tipos de usuario.</p>

	27. Habilitado solo para los tipos de usuario medico e internista.	
Nro. Colegiatura	29. Debe seleccionar un elemento. 31. Habilitado solo para los tipos de usuario medico e internista.	30. No debe seleccionar elemento. 32. Habilitado para todos los tipos de usuario.
RNE	33. Debe seleccionar un elemento. 35. Habilitado solo para los tipos de usuario medico e internista.	34. No debe seleccionar elemento. 36. Habilitado para todos los tipos de usuario.
Teléfono	37. No puede quedar en blanco. 39. Solo números. 41. Permite 7 o 9 dígitos	38. En blanco. 40. Ingresar letras 42. Permite menos de 7, diferente de 8 o más de 9 dígitos.
Contraseña	43. No puede quedar en blanco.	44. En blanco.

Fuente: Elaboración propia

En la tabla 91 se ve los campos del caso de uso gestionar usuario, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 92 Caso de Prueba Gestionar Usuario

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2,4,28,32,36	No selecciona tipo de usuario.	14/10/2109	No habilita campos.	Satisfactoria
2	8	Ingresa letras en Dni.	14/10/2109	No digita ningún texto.	Satisfactoria
3	10	Ingresa menos o más de 8 dígitos en Dni.	14/10/2109	Valida que se debe ingresar solo 8 dígitos o muestra mensaje de Dni incorrecto.	Satisfactoria
4	16	Selecciona fecha mayor a la actual.	14/10/2109	Muestra mensaje que no se puede seleccionar una fecha mayor a la actual.	Satisfactoria
5	22	No ingresa @ en el formato de correo.	14/10/2109	Muestra mensaje de completar correo electrónico.	Satisfactoria
6	40	Ingresa letras en Teléfono.	14/10/2109	No digita ningún texto.	Satisfactoria

7	42	Permite menos de 7, diferente de 8 o más de 9 dígitos.	14/10/2109	Valida que se debe ingresar solo 7 dígitos de teléfono o 9 dígitos para celular.	Satisfactoria
8	6	Clic en buscar	14/10/2109	Muestra mensaje de “No ha ingresado Dni”.	Satisfactoria
9	2 o 4 o 6 o 12 o 14 o 18 o 20 o 24 o 26 o 30 o 34 o 38 o 44	Ingresa o selecciona un dato, después de haber seleccionado tipo de usuario.	14/10/2109	Muestra mensajes de los campos a completar.	Satisfactoria
10	1,5,11,13,17,19,23,25,29,33,37,43	Completa campos correctamente y clic en registrar.	14/10/2109	Registrado correctamente.	Satisfactoria
11	1,5,11,13,17,19,23,25,29,33,37,43	Dni y datos existentes en BD.	14/10/2109	Muestra mensaje de Dni ya registrado anteriormente.	Satisfactoria

12	5	Clic en buscar.	14/10/2109	Carga data y desbloquea campos.	Satisfactoria
13	1,5,11,13,17,19,23,25,29,33,37,43	Actualiza data y clic en actualizar.	14/10/2109	Actualizado correctamente, luego bloquea campos.	Satisfactoria

Fuente: Elaboración propia

En la tabla 92 se puede ver todas las pruebas realizadas al caso de gestionar usuario, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 93 Prueba Funcional Gestionar Paciente

Condición /Evento / Campo	Clase válida	Clase no válida
DNI	1. No puede quedar en blanco. 3. Solo números. 5. Permite solo 8 dígitos	2. En blanco. 4. Ingresar letras 6. Permite 8 dígitos o mas
Nombre Completo	7. No puede quedar en blanco.	8. En blanco.
Sexo	9. Debe seleccionar un elemento.	10. No debe seleccionar elemento.
Fecha nacimiento	11. No puede quedar en blanco.	12. En blanco. 14. Selecciona fechas mayores a la actual.

	13. No puede seleccionar fechas mayores a la actual.	
Correo electrónico	15. No puede quedar en blanco. 17. Valida formato de correo.	16. En blanco. 18. Permite ingresar cualquier tipo de valor.
Estado paciente	19. Debe seleccionar un elemento.	20. No debe seleccionar elemento.
Teléfono	21. No puede quedar en blanco. 23. Solo números. 25. Permite 7 o 9 dígitos.	22. En blanco. 24. Ingresar letras. 26. Permite menos de 7, diferente de 8 o más de 9 dígitos.
Seguro	27. Debe seleccionar un elemento.	28. No debe seleccionar elemento.

Fuente: Elaboración propia

En la tabla 93 se ve los campos del caso de uso gestionar paciente, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 94 Caso de Prueba Gestionar Paciente

ID	Clase	Datos de entrada	Fecha	Resultado	Estado

1	4	Ingresa letras en Dni.	14/10/2109	No digita ningún texto.	Satisfactoria
2	6	Ingresa menos o más de 8 dígitos en Dni.	14/10/2109	Valida que se debe ingresar solo 8 dígitos o muestra mensaje de Dni incorrecto.	Satisfactoria
3	14	Selecciona fecha mayor a la actual.	14/10/2109	Muestra mensaje que no se puede seleccionar una fecha mayor a la actual.	Satisfactoria
4	18	No ingresa @ en el formato de correo.	14/10/2109	Muestra mensaje de completar correo electrónico.	Satisfactoria
5	24	Ingresa letras en Teléfono.	14/10/2109	No digita ningún texto.	Satisfactoria
6	26	Permite menos de 7, diferente de 8 o más de 9 dígitos.	14/10/2109	Valida que se debe ingresar solo 7 dígitos de teléfono o 9 dígitos para celular.	Satisfactoria

7	2	Clic en buscar.	14/10/2109	Muestra mensaje de “No ha ingresado Dni de Paciente”.	Satisfactoria
8	1	Clic en buscar.	14/10/2109	Carga data y desbloquea campos.	Satisfactoria
9	2 o 6 o 12 o 14 o 18 o 20 o 24 o 26 o 30 o 34 o 38 o 44	Ingresa o selecciona al menos un campo valido.	14/10/2109	Muestra mensajes de los campos a completar.	Satisfactoria
10	1,7,9,11,15,19,21,27	Completa campos correctamente y clic en registrar.	14/10/2109	Registrado correctamente.	Satisfactoria
11	1,7,9,11,15,19,21,27	Dni y datos existentes en BD.	14/10/2109	Muestra mensaje de Dni ya registrado anteriormente.	Satisfactoria
12	1,7,9,11,15,19,21,27	Actualiza data y clic en actualizar.	14/10/2109	Actualizado correctamente.	Satisfactoria

Fuente: Elaboración propia

En la tabla 94 se puede ver todas las pruebas realizadas al caso de gestionar paciente, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 95 Prueba Funcional Crear Orden Atención

Condición /Evento / Campo	Clase válida	Clase no válida
Dni	1. No puede quedar en blanco. 3. Solo números. 5. Permite solo 8 dígitos.	2. En blanco. 4. Ingresar letras. 6. Permite 8 dígitos o más.
Llegada paciente	7. Debe seleccionar un elemento.	8. No debe seleccionar elemento.
Detalle llegada paciente	9. No puede quedar en blanco.	10. En blanco.

Fuente: Elaboración propia

En la tabla 95 se ve los campos del caso de uso crear orden atención, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 96 Caso de Prueba Crear Orden Atención

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	4	Ingresa letras en Dni.	14/10/2109	No digita ningún texto.	Satisfactoria

2	6	Ingresa menos o más de 8 dígitos en Dni.	14/10/2109	Valida que se debe ingresar solo 8 dígitos o muestra mensaje de Dni incorrecto.	Satisfactoria
3	8 o 10	Ingresa o selecciona al menos un campo valido.	14/10/2109	Muestra mensajes de los campos a completar.	Satisfactoria
4	1,7,9	Completa campos correctamente y clic en crear orden.	14/10/2109	Registrado correctamente.	Satisfactoria

Fuente: Elaboración propia

En la tabla 96 se puede ver todas las pruebas realizadas al caso de crear orden atención, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 97 Prueba Funcional Listar Orden Atención

Condición /Evento / Campo	Clase válida	Clase no válida
Fecha inicio	1. Debe seleccionar una fecha menor o igual a la actual. 3. Debe seleccionar una fecha menor a la fecha fin	2. Debe seleccionar una fecha mayor a la actual. 4. Debe seleccionar una fecha mayor a la fecha fin

Fecha fin	5. Debe seleccionar una fecha menor o igual a la actual. 7. Debe seleccionar una fecha mayor a fecha inicio	6. Seleccionar una fecha mayor a la actual. 8. Debe seleccionar una fecha menor a fecha inicio
-----------	--	---

Fuente: Elaboración propia

En la tabla 97 se ve los campos del caso de uso listar orden atención, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 98 Caso de Prueba Listar Orden Atención

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2,4	Seleccionar fecha mayor a la actual.	17/10/2019	Muestra mensaje de alerta indicando que la fecha no debe ser mayor a la actual.	Satisfactoria
2	8	Seleccionar fecha fin menor a fecha inicio.	17/10/2019	Muestra mensaje de alerta indicando que la fecha fin debe ser mayor a la de fecha inicio.	Satisfactoria

3	6	Seleccionar fecha inicio mayor a fecha inicio.	17/10/2019	Muestra mensaje de alerta indicando que la fecha inicio debe ser menor a la de fecha fin.	Satisfactoria
4	1,3,5,7	Clic en buscar y fechas seleccionadas.	17/10/2019	Data cargada en las respectivas bandejas.	Satisfactoria

Fuente: Elaboración propia

En la tabla 98 se puede ver todas las pruebas realizadas al caso de listar orden atención, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 99 Prueba Funcional Listar Motivo Atención

Condición /Evento / Campo	Clase válida	Clase no válida
Motivo atención	1. Seleccionar un motivo atención.	2. No seleccionar un motivo atención.

Fuente: Elaboración propia

En la tabla 99 se ve los campos del caso de uso listar motivo atención, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 100 Caso de Prueba Listar Motivo Atención

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No seleccionar un motivo atención.	17/10/2019	Se cerró modal.	Satisfactoria
2	1	Seleccionar un motivo atención.	17/10/2019	Muestra pantalla de registro triaje.	Satisfactoria

Fuente: Elaboración propia

En la tabla 100 se puede ver todas las pruebas realizadas al caso de listar motivo atención, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 101 Prueba Funcional Registrar Triage

Condición /Evento / Campo	Clase válida	Clase no válida
Discriminador	1.Seleccionar discriminador.	2.No seleccionar discriminador.
Tópico	3. Seleccionar tópico.	4. No seleccionar tópico.
If (edad>14 && prioridad=1)	5. Campos signos vitales no obligatorios.	6. Campos signos vitales obligatorios.

If ((edad>14 && prioridad=2) (edad>14 && prioridad=3) (edad>14 && prioridad=4) (edad>14 && prioridad=5))	7. Campo PA obligatorio.	8. Campo PA no obligatorio.
If ((edad>14 && prioridad=2) (edad>14 && prioridad=3) (edad>14 && prioridad=4) (edad>14 && prioridad=5))	9. Campo T° obligatorio.	10. Campo T° no obligatorio.
If ((edad>14 && prioridad=2) (edad>14 && prioridad=3) (edad>14 && prioridad=4) (edad>14 && prioridad=5))	11. Campo SATO2 obligatorio.	12. Campo SATO2 no obligatorio.
If ((edad>14 && prioridad=2) (edad>14 && prioridad=3) (edad>14 && prioridad=4) (edad>14 && prioridad=5))	13. Campo FC obligatorio.	14. Campo FC no obligatorio.
If ((edad>14 && prioridad=2) (edad>14 && prioridad=3) (edad>14 && prioridad=4))	15. Campo FR obligatorio.	16. Campo FR no obligatorio.

(edad>14 && prioridad=5))		
If ((edad<=14 && prioridad=2) (edad<=14 && prioridad=3) (edad<=14 && prioridad=4) (edad<=14 && prioridad=5))	17. Campo Peso obligatorio.	18. Campo Peso no obligatorio.
If ((edad<=14 && prioridad=2) (edad<=14 && prioridad=3) (edad<=14 && prioridad=4) (edad<=14 && prioridad=5))	19. Campo T° obligatorio.	20. Campo T° no obligatorio.
If ((edad<=14 && prioridad=2) (edad<=14 && prioridad=3) (edad<=14 && prioridad=4) (edad<=14 && prioridad=5))	21. Campo SATO2 obligatorio.	22. Campo SATO2 no obligatorio.
If ((edad<=14 && prioridad=2) (edad<=14 && prioridad=3) (edad<=14 && prioridad=4) (edad<=14 && prioridad=5))	23. Campo FC obligatorio.	24. Campo FC no obligatorio.
If ((edad<=14 && prioridad=2) (edad<=14 && prioridad=3)	25. Campo FR obligatorio.	26. Campo FR no obligatorio.

(edad<=14 && prioridad=4) (edad<=14 && prioridad=5))		
If (edad<=14 && prioridad=1)	27. Campos signos vitales no obligatorios.	28. Campos signos vitales obligatorios.

Fuente: Elaboración propia

En la tabla 101 se ve los campos del caso de uso registrar triage, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 102 Caso de Prueba Registrar Triage

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No se selecciona discriminador.	18/10/2019	Muestra mensaje “Debe seleccionar un discriminador”.	Satisfactoria
2	3	No se selecciona tópico.	18/10/2019	Muestra mensaje “Debe seleccionar un tópico”.	Satisfactoria
3	8 o 10 o 12 o 14 o 16	No se registra campo con la condición de edad>14 y prioridad (2,3,4,5).	18/10/2019	Muestra mensajes con campos a completar.	Satisfactoria

4	18 o 20 o 22 o 24 o 26	No se registra campo con la condición de edad \leq 14 y prioridad (2,3,4,5,).	18/10/2019	Muestra mensajes con campos a completar.	Satisfactoria
5	6	Campos signos vitales obligatorios.	18/10/2019	Los campos no son restrictivos para registrar con esa condición.	Satisfactoria
6	28	Campos signos vitales obligatorios.	18/10/2019	Los campos no son restrictivos para registrar con esa condición.	Satisfactoria
7	1,3,7,9,11,13,15	Campos registrados.	18/10/2019	Triaje registrado correctamente.	Satisfactoria
8	1,3,17,19,21,23,25	Campos registrados.	18/10/2019	Triaje registrado correctamente.	Satisfactoria
9	1,3,5	Campos registrados.	18/10/2019	Triaje registrado correctamente.	Satisfactoria

10	1,3,27	Campos registrados.	18/10/2019	Triage registrado correctamente.	Satisfactoria
----	--------	---------------------	------------	----------------------------------	---------------

Fuente: Elaboración propia

En la tabla 102 se puede ver todas las pruebas realizadas al caso de registrar triage, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implementó con éxito.

Tabla 103 Prueba Funcional Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage

Condición /Evento / Campo	Clase válida	Clase no válida
Periodo	1. Seleccionar periodo.	2. No seleccionar periodo.
Excel	3. Selecciona botón Excel	4. No selecciona botón Excel

Fuente: Elaboración propia

En la tabla 103 se ve los campos del caso de uso generar reporte nivel porcentual de pacientes atendidos en triage, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 104 Caso de Prueba Generar Reporte Nivel Porcentual de Pacientes Atendidos en Triage

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No seleccionar periodo y	20/10/2019	Muestra mensaje “No ha	Satisfactoria

		clic en buscar		seleccionado periodo”	
2	1	Selecciono periodo y clic en buscar	20/10/2019	Carga data en bandeja	Satisfactoria
3	4	Selecciona botón Excel	20/10/2019	Exporta a formato Excel	Satisfactoria
4	3	No selecciona botón Excel	20/10/2019	No exporta a formato Excel	Satisfactoria

Fuente: Elaboración propia

En la tabla 104 se puede ver todas las pruebas realizadas al caso de generar reporte nivel porcentual de pacientes atendidos en triage, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 105 Prueba Funcional Generar Reporte Triage

Condición /Evento / Campo	Clase válida	Clase no válida
Periodo	5. Seleccionar periodo.	6. No seleccionar periodo.
Excel	7. Selecciona botón Excel	8. No selecciona botón Excel

Fuente: Elaboración propia

En la tabla 105 se ve los campos del caso de uso generar reporte triage, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 106 Caso de Prueba Generar Reporte Triage

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No seleccionar periodo y clic en buscar	20/10/2019	Muestra mensaje “No ha seleccionado periodo”	Satisfactoria
2	1	Selecciono periodo y clic en buscar	20/10/2019	Carga data en bandeja	Satisfactoria
3	4	Selecciona botón Excel	20/10/2019	Exporta a formato Excel	Satisfactoria
4	3	No selecciona botón Excel	20/10/2019	No exporta a formato Excel	Satisfactoria

Fuente: Elaboración propia

En la tabla 106 se puede ver todas las pruebas realizadas al caso de generar reporte triage, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 107 Prueba Funcional Gestionar Enfermedad Actual

Condición /Evento / Campo	Clase válida	Clase no válida
Motivo de ingreso	1. No puede quedar en blanco.	2. En blanco.
Forma de inicio	3. Debe seleccionar elemento.	4. No debe seleccionar elemento.
Curso de enfermedad	5. Debe seleccionar elemento.	6. No debe seleccionar elemento.
Tiempo de enfermedad	7. No puede quedar en blanco. 9. Debe seleccionar elemento.	8. En blanco. 10. No debe seleccionar elemento.
Relato cronológico	11. No puede quedar en blanco.	12. En blanco.

Fuente: Elaboración propia

En la tabla 107 se ve los campos del caso de uso gestionar enfermedad actual, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 108 Caso de Prueba Gestionar Enfermedad Actual

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
----	-------	------------------	-------	-----------	--------

1	2,8,12	No ingresa datos.	21/10/2019	Muestra mensaje de campos requeridos.	Satisfactoria
2	4,6,10	No selecciona campos.	21/10/2019	Muestra mensaje de campos requeridos.	Satisfactoria
3	1 o 3 o 5 o 7 o 9 o 11	Ingresa o selecciona datos, pero menos a uno.	21/10/2019	Muestra mensaje de campos requeridos.	Satisfactoria
4	1,3,5,7,9,11	Registra campos.	21/10/2019	Registrado correctamente.	Satisfactoria
5	1,3,5,7,9,11	carga campos después de registrar o modificar.	21/10/2019	carga correctamente.	Satisfactoria
6	1,3,5,7,9,11	modifica campos.	21/10/2019	Registrado correctamente.	Satisfactoria

Fuente: Elaboración propia

En la tabla 108 se puede ver todas las pruebas realizadas al caso de gestionar enfermedad actual, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 109 Prueba Funcional Gestionar Funciones Vitales

Condición /Evento / Campo	Clase válida	Clase no válida
Peso	1. No puede quedar en blanco.	2. En blanco.
talla	3. No puede quedar en blanco.	4. En blanco.

Fuente: Elaboración propia

En la tabla 109 se ve los campos del caso de uso gestionar funciones vitales, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 110 Caso de Prueba Gestionar Funciones Vitales

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2,4	No ingresa datos.	21/10/2019	Muestra mensaje de campos requeridos.	Satisfactoria
2	2 o 4	Ingresa campo menos uno	21/10/2019	Muestra mensaje de campos requeridos.	Satisfactoria
3	1,3	Calcula IMC	21/10/2019	Completa campo IMC	Satisfactoria

4	1,3	Registra campos.	21/10/2019	Registrado correctamente.	Satisfactoria
5	1,3	carga campos después de registrar o modificar.	21/10/2019	carga correctamente.	Satisfactoria
6	1,3	modifica campos.	21/10/2019	Registrado correctamente.	Satisfactoria

Fuente: Elaboración propia

En la tabla 110 se puede ver todas las pruebas realizadas al caso de gestionar funciones vitales, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 111 Prueba Funcional Gestionar Diagnóstico

Condición /Evento / Campo	Clase válida	Clase no válida
Modal diagnostico	1. Clic Agregar diagnóstico.	2. No agrega diagnóstico.
Buscar diagnostico	3. Busca diagnóstico.	4. No busca diagnóstico.
Añadir diagnostico	5. Clic icono añadir diagnóstico a la tabla.	6. No añade diagnostico a la tabla.

Fuente: Elaboración propia

En la tabla 111 se ve los campos del caso de uso gestionar diagnóstico, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 112 Caso de Prueba Gestionar Diagnóstico

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No selecciona botón agregar diagnóstico.	21/10/2019	No muestra pantalla flotante de agregar diagnóstico.	Satisfactoria
2	4	No busca diagnóstico de la lista de diagnósticos.	21/10/2019	Listado de diagnósticos cargado al aperturar pantalla flotante.	Satisfactoria
3	1	Selecciona botón agregar diagnóstico.	21/10/2019	Muestra pantalla flotante de listado de diagnósticos.	Satisfactoria
4	2	Busca diagnóstico de la lista de diagnósticos.	21/10/2019	Carga lista con diagnóstico consultado.	Satisfactoria
5	6	No añade datos de diagnóstico a tabla.	21/10/2019	Tabla de datos vacía.	Satisfactoria.

6	7	Añade datos de diagnóstico a tabla.	21/10/2019	Data cargada con datos de diagnóstico seleccionado.	Satisfactoria
7	5	Datos registrados	21/10/2019	Carga de datos registrados.	Satisfactoria

Fuente: Elaboración propia

En la tabla 112 se puede ver todas las pruebas realizadas al caso de gestionar diagnóstico, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 113 Prueba Funcional Gestionar Examen de Apoyo al Diagnostico

Condición /Evento / Campo	Clase válida	Clase no válida
Modal Examen de Apoyo al Diagnóstico	1. Clic agregar Examen de Apoyo al Diagnóstico.	2. No agrega Examen de Apoyo al Diagnóstico.
Buscar Examen de Apoyo al Diagnóstico	3. Busca Examen de Apoyo al Diagnóstico.	4. No busca Examen de Apoyo al Diagnóstico.
Añadir Examen de Apoyo al Diagnóstico	5. Clic icono añadir Examen de Apoyo al Diagnóstico a la tabla.	6. No añade Examen de Apoyo al Diagnóstico a la tabla.

Fuente: Elaboración propia

En la tabla 113 se ve los campos del caso de uso gestionar examen de apoyo al diagnóstico, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 114 Caso de Prueba Gestionar Examen de Apoyo al Diagnostico

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No selecciona botón agregar Examen.	21/10/2019	No muestra pantalla flotante de agregar Examen.	Satisfactoria
2	4	No busca Examen de Apoyo al Diagnóstico de la lista de Examen de Apoyo al Diagnóstico.	21/10/2019	Listado de Examen de Apoyo al Diagnóstico cargado al aperturar pantalla flotante.	Satisfactoria
3	1	Selecciona botón agregar Examen de Apoyo al Diagnóstico.	21/10/2019	Muestra pantalla flotante de listado de Examen de Apoyo al Diagnóstico.	Satisfactoria
4	2	Busca Examen de Apoyo al	21/10/2019	Carga lista con Examen de Apoyo al	Satisfactoria

		Diagnóstico de la lista de Examen de Apoyo al Diagnóstico.		Diagnóstico consultado.	
5	6	No añade datos de Examen de Apoyo al Diagnóstico a tabla.	21/10/2019	Tabla de datos vacía.	Satisfactoria.
6	7	Añade datos de Examen de Apoyo al Diagnóstico a tabla.	21/10/2019	Data cargada con datos de Examen de Apoyo al Diagnóstico seleccionado.	Satisfactoria
7	5	Datos registrados	21/10/2019	Carga de datos registrados.	Satisfactoria

Fuente: Elaboración propia

En la tabla 114 se puede ver todas las pruebas realizadas al caso de gestionar examen de apoyo al diagnóstico, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 115 Prueba Funcional Gestionar Receta

Condición /Evento / Campo	Clase válida	Clase no válida

Modal medicamento	1. Clic agregar medicamento.	2. No agrega medicamento.
Buscar medicamento	3. Busca medicamento.	4. No busca medicamento.
Añadir medicamento	5. Clic icono medicamento a la tabla.	6. No añade medicamento a la tabla.
Listar diagnostico	7.Lista diagnósticos	8.No lista diagnósticos.

Fuente: Elaboración propia

En la tabla 115 se ve los campos del caso de uso gestionar receta, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 116 Caso de Prueba Gestionar Receta

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No selecciona botón agregar medicamento.	21/10/2019	No muestra pantalla flotante de agregar medicamento.	Satisfactoria
2	4	No busca diagnóstico de la lista de medicamento.	21/10/2019	Listado de medicamento cargado al aperturar pantalla flotante.	Satisfactoria

3	1	Selecciona botón agregar medicamento.	21/10/2019	Muestra pantalla flotante de listado de medicamento.	Satisfactoria
4	2	Busca medicamento de la lista de medicamento.	21/10/2019	Carga lista con medicamento consultado.	Satisfactoria
5	6	No añade datos de medicamento a tabla.	21/10/2019	Tabla de datos vacía.	Satisfactoria.
6	7	Añade datos de medicamento a tabla.	21/10/2019	Data cargada con datos de medicamento seleccionado.	Satisfactoria
7	5	Datos registrados.	21/10/2019	Carga de datos registrados.	Satisfactoria
8	8	No listar diagnósticos.	21/10/2019	No lista diagnósticos al no tener datos en el formato de diagnóstico.	Satisfactoria
9	7	Listar diagnósticos.	21/10/2019	Cargar listado de diagnósticos	Satisfactoria

				al tener registro en formato diagnóstico.	
--	--	--	--	--	--

Fuente: Elaboración propia

En la tabla 116 se puede ver todas las pruebas realizadas al caso de gestionar receta, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 117 Prueba Funcional Gestionar Descanso Médico

Condición /Evento / Campo	Clase válida	Clase no válida
Del	1. Seleccionar fecha.	2. No seleccionar fecha.
Al	3. Seleccionar fecha.	4. No seleccionar fecha.
Observación	5. No puede quedar en blanco	6. En blanco
Listar diagnósticos	7. Lista diagnósticos	8. No lista diagnósticos.

Fuente: Elaboración propia

En la tabla 117 se ve los campos del caso de uso gestionar descanso médico, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 118 Caso de Prueba Gestionar Descanso Médico

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No selecciona fecha.	21/10/2019	Muestra campo a completar.	Satisfactoria
2	4	No selecciona fecha.	21/10/2019	Muestra campo a completar.	Satisfactoria
3	5	No registra campo.	21/10/2019	Muestra campo a completar.	Satisfactoria
4	1 o 3 o 5	Registra campos menos uno.	21/10/2019	Muestra campo a completar.	Satisfactoria
5	1,3	Seleccionar fecha “Del” mayor a fecha “Al”.	21/10/2019	Muestra mensaje de alerta que fecha “Del” no debe ser mayo a fecha “Al”.	Satisfactoria
6	8	No listar diagnósticos.	21/10/2019	No lista diagnósticos al no tener datos en el	Satisfactoria

				formato de diagnóstico.	
7	7	Listar diagnósticos.	21/10/2019	Cargar listado de diagnósticos al tener registro en formato diagnóstico.	Satisfactoria
8	1,3,5	Registra campos.	21/10/2019	Registro satisfactorio.	Satisfactoria

Fuente: Elaboración propia

En la tabla 118 se puede ver todas las pruebas realizadas al caso de gestionar descanso médico, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 119 Prueba Funcional Generar Reporte Nivel Percentil de Cumplimiento

Condición /Evento / Campo	Clase válida	Clase no válida
Periodo	9. Seleccionar periodo.	10. No seleccionar periodo.
Excel	11. Selecciona botón Excel	12. No selecciona botón Excel

Fuente: Elaboración propia

En la tabla 119 se ve los campos del caso de uso generar reporte nivel percentil de cumplimiento, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 120 Caso de Prueba Nivel Percentil de Cumplimiento

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No seleccionar periodo y clic en buscar	20/10/2019	Muestra mensaje “No ha seleccionado periodo”	Satisfactoria
2	1	Selecciono periodo y clic en buscar	20/10/2019	Carga data en bandeja	Satisfactoria
3	4	Selecciona botón Excel	20/10/2019	Exporta a formato Excel	Satisfactoria
4	3	No selecciona botón Excel	20/10/2019	No exporta a formato Excel	Satisfactoria

Fuente: Elaboración propia

En la tabla 120 se puede ver todas las pruebas realizadas al caso de generar reporte nivel percentil de cumplimiento, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

**Tabla 121 Prueba Funcional Generar Reporte
Emergencia**

Condición /Evento / Campo	Clase válida	Clase no válida
Periodo	13. Seleccionar periodo.	14. No seleccionar periodo.
Excel	15. Selecciona botón Excel	16. No selecciona botón Excel

Fuente: Elaboración propia

En la tabla 121 se ve los campos del caso de uso generar reporte emergencia, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

**Tabla 122 Caso de Prueba Generar Reporte
Emergencia**

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No seleccionar periodo y clic en buscar	20/10/2019	Muestra mensaje “No ha seleccionado periodo”	Satisfactoria
2	1	Selecciono periodo y clic en buscar	20/10/2019	Carga data en bandeja	Satisfactoria
3	4	Selecciona botón Excel	20/10/2019	Exporta a formato Excel	Satisfactoria

4	3	No selecciona botón Excel	20/10/2019	No exporta a formato Excel	Satisfactoria
---	---	---------------------------	------------	----------------------------	---------------

Fuente: Elaboración propia

En la tabla 122 se puede ver todas las pruebas realizadas al caso de generar reporte emergencia, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 123 Prueba Funcional Gestionar Nota Enfermería

Condición /Evento / Campo	Clase válida	Clase no válida
Nota Enfermería	1. No puede quedar en blanco.	2. En blanco.
Listar diagnósticos.	3. Lista diagnósticos	4. No lista diagnósticos.

Fuente: Elaboración propia

En la tabla 123 se ve los campos del caso de uso gestionar nota enfermería, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 124 Caso de Prueba Gestionar Nota Enfermería

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	2	No se registra campo notas enfermería.	20/10/2019	Muestra mensaje de campo requerido.	Satisfactoria.

2	4	No listar diagnósticos.	21/10/2019	No lista diagnósticos al no tener datos en el formato de diagnóstico.	Satisfactoria
3	3	Listar diagnósticos.	21/10/2019	Cargar listado de diagnósticos al tener registro en formato diagnóstico.	Satisfactoria
4	1	Registra notas enfermería.	21/10/2019	Registro satisfactorio.	Satisfactoria

Fuente: Elaboración propia

En la tabla 124 se puede ver todas las pruebas realizadas al caso de gestionar nota enfermería, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 125 Prueba Funcional Actualizar Abandono

Condición /Evento / Campo	Clase válida	Clase no válida
Abandono	1. Botón abandono activo en los estados pendiente y en atención en las bandejas internista, médico y enfermera.	2. Botón abandono activo en el estado atendido en las bandejas internista, médico y enfermera.

	3. Botón visible en las bandejas de internista, médico y enfermera.	4. Botón visible en la bandeja de admisionista.
--	---	---

Fuente: Elaboración propia

En la tabla 125 se ve los campos del caso de uso actualizar abandono, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

**Tabla 126 Caso de Prueba Actualizar abandono
Abandono**

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	4	Botón abandono visible al seleccionar fila en la bandeja de admisionista.	17/10/2019	No se visualizó botón abandono en la bandeja admisionista.	Satisfactoria
2	3	Botón abandono visible al seleccionar fila en las bandejas de internista médico y enfermera.	17/10/2019	Se visualizó botón abandono en las respectivas bandejas.	Satisfactoria

3	2	Botón abandono activo en el estado atendido al seleccionar fila en las bandejas internista, médico y enfermera.	18/10/2019	No se activó botón al seleccionar la fila de la bandeja.	Satisfactoria
4	1,3	Botón abandono activo en los estados pendiente y en atención al seleccionar fila en las bandejas internista, médico y enfermera.	18/10/2019	Se activó botón al seleccionar fila.	Satisfactoria
5	1,3	Actualizar estado abandono.	18/10/2019	Se actualizo correctamente.	Satisfactoria

Fuente: Elaboración propia

En la tabla 126 se puede ver todas las pruebas realizadas al caso de actualizar abandono actual, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 127 Prueba Funcional Actualizar Alta Medica

Condición /Evento / Campo	Clase válida	Clase no válida
Alta medica	<p>1. Botón alta médica activo en el estado en atención en la bandeja médico.</p> <p>3.Botón visible solo en la bandeja médico.</p>	<p>2.Botón alta médica activo en los estados pendiente y atendido en la bandeja médico.</p> <p>4.Botón visible en las bandejas admisionista, internista y enfermera.</p>

Fuente: Elaboración propia

En la tabla 127 se ve los campos del caso de uso actualizar alta médica, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 128 Caso de Prueba Actualizar Alta Medica

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	4	Botón alta médica visible al seleccionar fila en las bandejas admisionista,	17/10/2019	No se visualizó botón alta médica en las bandejas.	Satisfactoria

		internista y enfermera.			
2	3	Botón alta médica visible solo al seleccionar fila en la bandeja médico.	17/10/2019	Se visualizó botón alta médica en la bandeja médico.	Satisfactoria
3	2	Botón alta médica activo en los estados pendiente y atendido al seleccionar fila en la bandeja médico.	18/10/2019	No se activó botón al seleccionar la fila de la bandeja.	Satisfactoria
4	1,3	Botón alta médica activo en el estado en atención al seleccionar fila en la bandeja médico.	18/10/2019	Se activó botón al seleccionar fila.	Satisfactoria

5	1,3	Actualizar alta médica con diagnóstico y notas de enfermería registrada.	18/10/2019	Se actualizo correctamente.	Satisfactoria
6	1,3	Actualizar alta médica sin diagnóstico o notas de enfermería registrada.	18/10/2019	Muestra mensaje de “no posee diagnóstico o nota de enfermería”.	Satisfactoria

Fuente: Elaboración propia

En la tabla 128 se puede ver todas las pruebas realizadas al caso de actualizar alta médica, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 129 Prueba Funcional Reporte Atención Emergencia

Condición /Evento / Campo	Clase válida	Clase no válida
Reporte	<p>1.Botón reporte activo en el estado atendido en la bandeja médico.</p> <p>3.Botón visible solo en la bandeja médico.</p>	<p>2.Botón reporte activo en el estado pendiente y en atención en la bandeja médico.</p> <p>4.Botón visible en las bandejas internista, admisionista y enfermera.</p>

Fuente: Elaboración propia

En la tabla 129 se ve los campos del caso de uso reporte atención emergencia, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 130 Caso de Prueba Reporte Atención Emergencia

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
1	4	Botón reporte visible al seleccionar fila en las bandejas internista, admisionista y enfermera.	17/10/2019	No se visualizó botón alta médica en las bandejas.	Satisfactoria
2	3	Botón reporte visible al seleccionar fila solo en la bandeja médico.	17/10/2019	Se visualizó botón alta médica en la bandeja médico.	Satisfactoria
3	2	Botón reporte activo en el estado pendiente y en atención al seleccionar fila en la bandeja médico.	18/10/2019	No se activó botón al seleccionar la fila de la bandeja.	Satisfactoria

4	1	Botón reporte activo en el estado atendido al seleccionar fila en la bandeja médico.	18/10/2019	Se activó botón al seleccionar fila.	Satisfactoria
5	1,3	Generar reporte atención emergencia.	18/10/2019	Se exporto reporte a formato pdf correctamente.	Satisfactoria

Fuente: Elaboración propia

En la tabla 130 se puede ver todas las pruebas realizadas al caso de reporte atención emergencia, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.

Tabla 131 Prueba Funcional Actualizar Acto Enfermería

Condición /Evento / Campo	Clase válida	Clase no válida
Acto enfermería	1.Botón acto enfermería activo en el estado en atención en la bandeja médico.	2.Botón acto enfermería activo en el estado pendiente y atendido en la bandeja médico.

	3.Botón visible solo en la bandeja enfermera.	4.Botón visible en las bandejas internista, admisionista y médico.
--	---	--

Fuente: Elaboración propia

En la tabla 131 se ve los campos del caso de uso actualizar acto enfermería, así como las clases validas (criterios aceptados por el sistema) y clases no validas (criterios no aceptados por el sistema), estos permitieron realizar las pruebas funcionales.

Tabla 132 Caso de Prueba Acto Enfermería

ID	Clase	Datos de entrada	Fecha	Resultado	Estado
12	4	Botón acto enfermería visible al seleccionar fila en las bandejas internista, admisionista y médico.	17/10/2019	No se visualizó botón alta médica en las bandejas.	Satisfactoria
13	3	Botón acto enfermería visible al seleccionar fila solo en la bandeja enfermera.	17/10/2019	Se visualizó acto enfermería en la bandeja enfermera.	Satisfactoria

20	2	Botón acto enfermería activo en el estado pendiente y atendido al seleccionar fila en la bandeja médico.	18/10/2019	No se activó botón al seleccionar la fila de la bandeja.	Satisfactoria
21	1	Botón acto enfermería activo en el estado en atención al seleccionar fila en la bandeja médico.	18/10/2019	Se activó botón al seleccionar fila.	Satisfactoria
25	1,3	Actualizar acto enfermería con nota de enfermería registrada.	18/10/2019	Se actualizo correctamente.	Satisfactoria
27	1,3	Actualizar acto enfermería sin nota de	18/10/2019	Muestra mensaje de “no posee nota de	Satisfactoria

		enfermería registrada.		enfermería registrada”.	
--	--	------------------------	--	-------------------------	--

Fuente: Elaboración propia

En la tabla 132 se puede ver todas las pruebas realizadas al caso de actualizar acto enfermería, en donde se obtuvo los resultados esperados, demostrando de esta manera que la funcionalidad se desarrolló e implemento con éxito.